



Fakultät für Ingenieurwissenschaften, Informatik und Psychologie
Institut für Datenbanken und Informationssysteme
Leiter: Prof. Dr. Manfred Reichert

Dissertation

zur Erlangung des Doktorgrades Dr. rer. nat.
der Fakultät für Ingenieurwissenschaften, Informatik und Psychologie
der Universität Ulm

Robuste und kontextbezogene Ausführung mobiler Aktivitäten in Prozessumgebungen

vorgelegt von

Rüdiger Christoph Pryss

aus Illertissen

2015

Amtierende Dekanin

Prof. Dr. Tina Seufert

Gutachter

Prof. Dr. Manfred Reichert
Prof. Dr. Franz J. Hauck

Tag der Promotion

05.10.2015

für meinen Vater..

Vorwort

Die vorliegende Arbeit entstand während meiner Tätigkeit als Doktorand in der Abteilung Datenbanken und Informationssysteme der Universität Ulm. Die Problemstellungen dieser Arbeit wurden im Rahmen des Forschungsprojekts Marple (*Managing Robust Mobile Processes in a Complex World*) identifiziert und bearbeitet. Sie war darüber hinaus durch Teilprojekte mit externen Partnern auch durch praktische Fragestellungen geprägt. Als Ergebnis liegt mit dieser Arbeit ein umfassendes Rahmenwerk vor, dessen Konzepte die robuste und kontextbezogene Ausführung mobiler Aktivitäten in Prozessumgebungen bewerkstelligen. An dieser Stelle möchte ich mich bei allen Personen bedanken, ohne deren Unterstützung diese Arbeit so nicht möglich gewesen wäre.

Mein herzlicher Dank gilt Prof. Dr. Manfred Reichert für die Betreuung dieser Arbeit, für das Korrekturlesen aller Schriftstücke während meiner Promotionszeit, die vielen Diskussionen, seine freundschaftliche Unterstützung und seinen unermüdlichen Einsatz als Doktorvater.

Ich danke Herrn Prof. Dr. Franz J. Hauck für die Übernahme des Koreferats und seinem Interesse an dem Thema. Ebenfalls danken möchte ich Herrn Prof. Dr. Peter Dadam für das Interesse am Thema, seine Unterstützung und die konstruktiven Diskussionen, die mich stets voranbrachten.

Während meiner Promotionszeit sind neue Freundschaften entstanden, die meine Zeit am Institut geprägt haben. So sind Freundschaften zu Marc Schickler, Dr. Winfried Schlee, Johannes Schobel, Dr. Stephan Buchwald, Dr. Vera Künzle, Dr. Jens Kolb, Dr. Dominic Müller, Andreas Lanz, Dr. Ulrich Kreher, Michael Predeschly und Julian Tiedeken entstanden.

Auch möchte ich meinen Kollegen David Knuplesch, Nicolas Mundbrod und Dr. Thao Ly herzlich danken, mit denen mich eine intensive Zeit verbindet. Des Weiteren möchte ich mich bei Carolina Chiao, Jochen Herrmann, Philip Geiger, Steffen Musiol, Dominik Deuter, Martin Liebrecht, Michael Stach und David Langer für ihre Unterstützung bedanken.

Ich möchte mich aus der Fakultät für Informatik der Universität Ulm bei Prof. Dr. Heiko Neumann und Dr. Klaus Murmann bedanken, die mit ihrer hilfsbereiten Art und Unterstützung meine Promotionszeit mitgeprägt haben.

Außerdem wurde meine Promotionszeit von den beiden befreundeten Arbeitsgruppen aus Konstanz und Regensburg geprägt. So danke ich Prof. Dr. Thomas Elbert und seiner Arbeitsgruppe aus Konstanz für die gemeinsame Arbeit. In Regensburg danke ich Herrn Dr. Winfried Schlee und PD Dr. Berthold Langguth und ihrer Arbeitsgruppe für die gemeinsame Arbeit.

Ich möchte mich bei meiner Mutter, meinem Vater, meiner Schwester und Markus von ganzem Herzen für ihre Geduld, ihr Verständnis und ihren Rückhalt bedanken. Ohne sie wäre ich sicher nicht so weit gekommen. Zu guter Letzt möchte ich mich bei meinen Freunden Simon Mayer, Dr. Bernhard Miller, Stefan Briel, Dr. Ralph Bobrik, Stefan Huber, Hartwig Bentele, Dr. Werner Schebesta und Dr. Dieter Reinhardt für ihre Geduld und Unterstützung bedanken.

Kurzfassung

IT-Trendanalysten sehen das Thema "Mobilität" als eine wichtige Säule nachhaltiger IT-Lösungen. Der Trend in Richtung mobiler IT-Anwendungen wird maßgeblich durch Millenials getrieben, d.h. Menschen die mit dem digitalen Zeitalter aufgewachsen sind. Diese erwarten insbesondere auch eine Integration von Smart-Mobilgeräten in bestehende IT-Lösungen. In Bezug auf Prozess-Management-Technologie bedeutet dieser Trend, dass Smart-Mobilgeräte in IT-gestützte Arbeits- bzw. Prozessabläufe nahtlos integriert werden können müssen. Insbesondere sollten sowohl einzelne Aktivitäten (d.h. Prozessschritte) als auch ganze Prozessfragmente (d.h. Ausschnitte eines Prozesses) auf Smart-Mobilgeräten ausführbar sein.

Die vorliegende Arbeit adressiert eine solche Integration von Prozess-Management-Technologie und Smart-Mobilgeräten. Konkret wird untersucht, wie ausgewählte Aktivitäten eines Prozesses robust und kontextbezogen auf Smart-Mobilgeräten ausgeführt werden können und welche weitergehenden Anforderungen sich für mobil ausgeführte Aktivitäten im Vergleich zur Ausführung von Aktivitäten auf stationären Systemen ergeben. Da Smart-Mobilgeräte beschränkte Ressourcen besitzen und das Risiko eines Ausfalls höher als bei stationären Systemen ist, erfordern diese Aspekte tiefergehende Untersuchungen. Darüber hinaus erfordert die Unterstützung mobiler Aktivitäten eine technische Umgebung, in der Prozesse ausgeführt werden (sog. Prozessumgebung).

Die Arbeit zeigt, dass die nahtlose Integration von Smart-Mobilgeräten in eine Prozessumgebung einen mobilen Kontext (d.h. Attribute wie z.B. Ausführungsort, Geräteeigenschaften und Netzverbindung) erfordert. Auf dessen Basis wird ein umfassendes Rahmenwerk eingeführt, mit dem sich mobile Aktivitäten robust und kontextbezogen in einer Prozessumgebung ausführen lassen. Das Rahmenwerk fußt auf fünf technischen Säulen, deren Konzepte die robuste und kontextbezogene Ausführung bewerkstelligen. Darüber hinaus wird gezeigt, wie sich die vorgestellte Lösung in existierende Prozess-Management-Technologie integrieren lässt. Insgesamt eröffnet eine robuste und kontextbezogene Ausführung mobiler Aktivitäten in einer Prozessumgebung neue Perspektiven für die Einbindung von Endanwendern in ihre Prozesse.

Inhaltsverzeichnis

I	Problembeschreibung und Anforderungsanalyse	3
1	Einleitung	5
1.1	Hintergrund: Enterprise Mobility Management	6
1.2	Beitrag	7
1.3	Methodik	10
1.4	Aufbau	10
2	Anforderungsanalyse	13
2.1	Literaturrecherche	13
2.1.1	Literaturrecherchephase 1: Suchmaschinen	14
2.1.2	Literaturrecherchephase 2: Clusteranalyse	15
2.1.3	Ergebnis der Literaturrecherche	16
2.2	Fallstudien	17
2.2.1	Klinische Visite	17
2.2.2	Luftfahrt-Catering	21
2.2.3	Telecare	24
2.3	Anforderungsanalyse	25
2.3.1	Funktionale Anforderungen mobiler Aktivitäten	25
2.3.2	Funktionale Anforderungen des mobilen Kontexts	27
2.3.3	Nicht-funktionale Anforderungen mobiler Aktivitäten	27
2.4	Zusammenfassung	28
3	Mobile Prozesse	29
3.1	Einleitung	29
3.2	Prozessumgebung	30
3.3	Sichtweisen auf mobile Prozesse	31
3.3.1	Sichtweise 1: Betrachtung des Aktivitätentyps	31
3.3.2	Sichtweise 2: Betrachtung des Klient-Server-Modells	33
3.3.3	Sichtweise 3: Betrachtung des mobilen Prozess-Klienten	36
3.3.4	Sichtweise 4: Betrachtung von Ad-hoc Änderungen	41
3.4	Diskussion	42
3.5	Zusammenfassung	44
4	Stand der Technik	45
4.1	Forschungsprojekte	45
4.2	Ausgewählte Forschungsansätze	48
4.3	Zuteilung von Endanwendern zu Aktivitäten	49
4.4	Prozess-Constraints	50
4.5	Mobiler Kontext	50
4.6	Existierende Technologien	51
4.7	Diskussion	52

4.8	Zusammenfassung	54
II	Prozessumgebung	55
5	Prozess-Metamodell	57
5.1	Einleitung	57
5.2	Prozess-Lebenszyklus	58
5.3	Prozess-Metamodell	59
5.4	Endanwendermodell	65
5.4.1	Modellierzeit	65
5.4.2	Ausführungszeit	67
5.5	Zusammenfassung	69
6	Prozess-Architektur	71
6.1	Einleitung	71
6.2	Architekturkomponenten	72
6.3	Prozess-Klient	73
6.4	Worklists	74
6.4.1	Worklist-Einträge	75
6.4.2	Worklist-Operationen	76
6.4.3	Worklist-Befüllung	76
6.5	Protokoll der Aktivitätsausführung	77
6.6	Zusammenfassung	79
III	Technische Lösung	81
7	Mobiler Kontext	83
7.1	Einleitung	84
7.2	Forschungsmethodik	86
7.3	Parameterkatalog	90
7.3.1	Parameterkategorie I: Smart-Mobilgerät	90
7.3.2	Parameterkategorie II: Mobile Aktivität	93
7.3.3	Parameterkategorie III: Gesamtprozess	97
7.3.4	Parameterkategorie IV: Mobiler Endanwender	98
7.3.5	Sonderfälle	99
7.4	Weitere Aspekte	100
7.5	Diskussion	100
7.5.1	Eignung des mobilen Kontexts	101
7.5.2	Spezifikations- und Verwaltungsaufwand	102
7.6	Zusammenfassung	103
8	Modellierung, Erzeugung und Ausführung mobiler Aktivitäten	105
8.1	Einleitung	105
8.2	Modellierung mobiler Aktivitäten	106
8.3	Erzeugung von Ausführungsinstanzen mobiler Aktivitäten	110

8.3.1	mobiler Kontext	110
8.3.2	Ranking-Algorithmus	111
8.3.3	mobiler Kontext und Worklist-Sortierung	113
8.4	Ausführung mobiler Aktivitäten	115
8.4.1	Systemwechsel	117
8.5	Weitere Aspekte	118
8.6	Diskussion	121
8.6.1	Beitrag zur robusten Ausführung mobiler Aktivitäten	121
8.6.2	Verwandte Ansätze	121
8.7	Zusammenfassung	122
9	 Mobile Aktivitäten und Ausnahmebehandlung	123
9.1	Einleitung	123
9.2	Forschungsmethodik	124
9.3	Ausnahmebehandlungsstrategie mobiler Aktivitäten	125
9.4	Delegation mobiler Aktivitäten	126
9.4.1	Fragestellung DA1: Starten einer Delegation	127
9.4.2	Fragestellung DA2: Ermittlung geeigneter mobiler Endanwender	128
9.4.3	Fragestellung DA3: Delegationsdurchführung	128
9.4.4	Fragestellung DA4: Wichtige Aspekte für den Endanwender	130
9.4.5	Fragestellung DA5: Wichtige Aspekte für das PrMS	132
9.4.6	Fragestellung DA6: Delegationsvarianten	135
9.5	Abbrechen mobiler Aktivitäten	139
9.6	Backup mobiler Aktivitäten	140
9.6.1	Fragestellung BA1: Grundgedanke	140
9.6.2	Fragestellung BA2: Backupvarianten	140
9.6.3	Fragestellung BA3: Wichtige Aspekte für den Endanwender	142
9.6.4	Fragestellung BA4: Wichtige Aspekte für das PrMS	143
9.7	Weitere Aspekte	144
9.7.1	Offline-Aktivitäten	145
9.7.2	Parametergewichtung bei Delegationen	146
9.7.3	Übergabe von Aktivitäten	146
9.7.4	Ausführungsprotokoll und Parameterübertragung	146
9.7.5	Ausführungshistorie	147
9.8	Diskussion	149
9.8.1	Beitrag zur robusten Ausführung mobiler Aktivitäten	149
9.8.2	Verwandte Ansätze	149
9.9	Zusammenfassung	150
10	 Mobile Aktivitäten und Prozess-Constraints	151
10.1	Einleitung	151
10.2	Praxisrelevanz	152
10.3	Anpassungen bei der Erzeugung und Ausführung mobiler Aktivitäten	152
10.3.1	Anpassung 1: Zuteilungsschema BoD-Aktivität A1	153
10.3.2	Anpassung 2: Worklist-Information zu abhängiger BoD-Aktivität A2	156
10.3.3	Anpassung 3: Priorisierung BoD-Aktivität A1	156
10.3.4	Anpassung 4: Worklist-Information zu BoD-Zwischenaktivitäten	157

10.3.5	Anpassung 5: Zuteilungsschema Card-Aktivität	159
10.3.6	Anpassungen 6-7: Worklist-Priorisierung von Card-Aktivitäten	160
10.3.7	Anpassungen bei kombinierter Anwendung der Prozess-Constraints	160
10.4	Anpassungen bei der Ausnahmebehandlung mobiler Aktivitäten	161
10.4.1	Anpassung 8: Zuteilungsschema SoD-Aktivität A1	161
10.4.2	Anpassung 9: Zuteilungsschema Backup SoD-Aktivität A2	164
10.4.3	Anpassung 10: Zuteilungsschema Delegation BoD-Aktivität A1	164
10.4.4	Anpassung 11: Ausnahmebehandlung BoD-Aktivität A1	164
10.4.5	Anpassung 12: Ausnahmebehandlung BoD-Aktivität A2	165
10.4.6	Anpassung 13: Ausnahmebehandlung Card-Aktivität	165
10.4.7	Anpassung 14: Ausnahmebehandlung Offline-Aktivitäten	166
10.4.8	Anpassungen bei kombinierter Anwendung der Prozess-Constraints	166
10.5	Weitere Aspekte	166
10.5.1	Berücksichtigung weiterer Prozess-Constraints	166
10.5.2	Inkonsistente Constraint-Modellierung	168
10.6	Diskussion	168
10.6.1	Beitrag zur robusten Ausführung mobiler Aktivitäten	168
10.6.2	Verwandte Ansätze	169
10.7	Zusammenfassung	170
11	Architektur Aspekte	175
11.1	Einleitung	175
11.2	Mobiler Prozess-Klient	177
11.2.1	Benutzerschnittstelle	177
11.2.2	Endanwender-Manager	178
11.2.3	Geräte-Manager	178
11.2.4	Aktivitäten-Manager	178
11.2.5	Weitere Komponenten	179
11.3	Ausführungsengine	179
11.3.1	Endanwender-Manager	179
11.3.2	Geräte-Manager	179
11.3.3	Ausführungsengine	181
11.4	Prozess-Modellierung	183
11.5	Informationsarchitektur des mobilen Prozess-Klienten	184
11.6	Verwandte Ansätze	185
11.7	Zusammenfassung	186
IV	Validation der Konzepte	187
12	Prototypische Realisierung	189
12.1	Einleitung	189
12.2	Service-orientierte Middleware	190
12.3	Schnittstelle zum Prozess-Management-System	192
12.4	Schnittstelle zum mobilen Prozess-Klienten	193
12.5	Kernel	193
12.5.1	Zuteilung mobiler Aktivitäten	194

12.5.2	Ausführung mobiler Aktivitäten	194
12.5.3	Beenden mobiler Aktivitäten	195
12.6	Repository	195
12.7	Mobiler Prozess-Klient	196
12.8	Weitere Aspekte	198
12.9	Zusammenfassung	199
13	Praktische Anwendung	201
13.1	Einleitung	201
13.2	Fallstudie 1: Lagerverwaltung	201
13.3	Fallstudie 2: Rettungsdienst	202
13.4	Fallstudie 3: Mobile Datenerfassung in Studien	204
13.5	Diskussion	205
13.6	Zusammenfassung	206
V	Fazit	207
14	Zusammenfassung und Ausblick	209
Anhang		212

Teil I

Problembeschreibung und Anforderungsanalyse

Wir warten unser Leben lang auf den außergewöhnlichen Menschen, statt die gewöhnlichen um uns her in solche zu verwandeln.

Hans Urs von Balthasar (1905-1988)

1

Einleitung

Der Begriff *Millennials* spiegelt die voranschreitende Digitalisierung unseres Alltags wider—*Millennials* bezeichnen Menschen, die mit dem digitalen Zeitalter, insbesondere der digitalen Kommunikation, aufgewachsen sind [MQ114, HS07]. Sie stellen daher an ihre Arbeits- und Privatwelt neuartige Anforderungen und haben auch konkrete Vorstellungen dazu, was ihnen Informationstechnologie (IT) bieten soll. Einerseits eröffnet diese Entwicklung viele Chancen und Geschäftsfelder, andererseits ergeben sich große Herausforderungen. Das Bundesministerium für Wirtschaft und Energie spricht in diesem Kontext von smarten Diensten (engl. *Smart Services*), die von der Industrie geschaffen und angeboten werden sollen, um auch in Zukunft konkurrenzfähig zu bleiben. Moderne IT hält daher zunehmend Einzug in zahlreiche Bereiche unseres Alltags, um unterschiedlichste Dienste *smarter* zu gestalten. Leider resultieren hieraus aber oftmals isolierte IT-Lösungen, die zwar einzelne Bereiche *smarter* machen, sich aber nicht in eine anspruchsvolle Gesamtlösung integrieren lassen. Dies wiederum hat die Erkenntnis reifen lassen, dass viele *smarte* Dienste auf mittlere Sicht zu teuer werden und nicht den erhofften Nutzen bringen.

Die Menge der Geschäftsfelder, die von den Anbietern *smarter* Dienste adressiert werden, ist groß. Sie umfasst zum Beispiel *Mobile Device Management*, *Enterprise Mobility Management*, *Cloud Computing*, *Big Data*, *Unified Communication & Collaboration*, *Social Media*, *Industrie 4.0*, *Augmented Reality*, *3D-Druck* oder *Touch-Technologie*. Die hier entstandenen Technologien nahtlos in eine Gesamtlösung zu integrieren, ist eine komplexe Aufgabe, die Unternehmen vor große Herausforderungen stellt.

Entlang dieser Trends haben sich einige Integrationsansätze in den vergangenen Jahren positiv entwickelt. Einen besonders wichtigen Beitrag liefert Prozess-Management-Technologie (*PrMT*) [RW12, Wes07, AVH04], mittels der es viele Unternehmen geschafft haben, ihre Geschäftsprozesse agiler, flexibler und kostengünstiger zu gestalten. Allerdings beklagen viele Unternehmen im Kontext *smarter* Dienste deren fehlende bzw. unzureichende Verknüpfung mit *PrMT*. Dies trifft auch für das *Enterprise Mobility Management* zu.

Smart-Mobilgeräte (z.B. Smartphones und Tablets) als Kernkomponente des *Enterprise Mobility Managements* haben sowohl bezogen auf ihre Hardwareeigenschaften als auch ihre mobilen Betriebssysteme eine beispiellose technische Entwicklung durchlaufen. Viele Möglichkeiten, die bisher nur Desktop-Systemen vorbehalten waren, sind in den letzten Jahren auf jedem handelsüblichen Smart-Mobilgerät umgesetzt worden.

Die Integration von Smart-Mobilgeräten mit *PrMT* hat dagegen bislang nur wenig Beachtung gefunden. Dass die Entwicklung einer solchen Verknüpfung erst am Anfang steht und zudem komplex ist, wird in dieser Arbeit anhand verschiedener Konzepte zur Integration von Smart-Mobilgeräten mit *PrMT* verdeutlicht. Ziel ist es, dass *Millennials* bei den alltäglichen Aufgaben ihrer Arbeits- und Privatwelt durch geeignete Verknüpfung von *PrMT* mit Smart-Mobilgeräten unterstützt werden. Im Fokus steht dabei insbesondere die robuste mobile Unterstützung ausgewählter Aktivitäten eines Geschäftsprozesses. Trotz zunehmend besserer Eigenschaften von Smart-Mobilgeräten, müssen dabei noch zahlreiche Herausforderungen, etwa Verbindungsabbrüche oder niedrige Akkustände, berücksichtigt werden.

1.1 Hintergrund: Enterprise Mobility Management

Die angemessene Berücksichtigung von Mobilität in IT-Systemen stellt sich äußerst komplex dar. So wird im Gartner¹-Bericht *The Mobile Imperative: Getting Started and Overcoming Obstacles* [GAR14d] diskutiert, welche Hindernisse sich dem Paradigma des *Enterprise Mobility Managements* in den Weg stellen; [GAR14a] wiederum nennt Faktoren, die erfüllt sein müssen, damit ein Unternehmen für das Paradigma des *Enterprise Mobility Managements* bereit ist.

Worin die speziellen Herausforderungen des *Enterprise Mobility Managements* liegen, zeigt folgendes Zitat [GAR14d]: "While most enterprises are increasingly feeling the imperative to do mobile, many don't know where to begin, and there are many obstacles to success. These range from development to deployment, and span issues ranging from tools to management and security, and increasingly take their cues from consumerization and bring your own device (BY-OD)." Der erste Teil dieses Zitats verdeutlicht, dass viele Unternehmen den Trend *Enterprise Mobility Management* für sich als wichtig erkannt haben. Allerdings wird auch ausgedrückt, dass viele Unternehmen nicht wissen, womit sie genau beginnen sollen und welche Hindernisse tatsächlich über Erfolg bzw. Misserfolg einer mobilen Lösung entscheiden. Schließlich muss sich ein Unternehmen die Frage stellen, wie Anwendungen für das *Enterprise Mobility Managements* entwickelt werden sollen, wie sie in Betrieb genommen werden können und welche Werkzeuge hierbei zur Anwendung kommen sollen. Daneben darf der Aspekt der Sicherheit nicht vernachlässigt werden. Der letzte Teil des Zitats spricht einen Aspekt an, der aktuell ebenfalls intensiv diskutiert wird, das sog. BYOD²-Prinzip. Im Kern geht es um die Frage, wie private Smart-Mobilgeräte von Mitarbeitern in das *Enterprise Mobility Management* integriert werden können. Für Unternehmen verspricht eine solche Integration enorme Einsparpotentiale, da nahezu jeder Mitarbeiter ein privates Smart-Mobilgerät besitzt, das prinzipiell auch für geschäftliche Zwecke genutzt werden kann. Jedoch wirft dies neue Herausforderungen auf, etwa die sichere Integration der privaten Smart-Mobilgeräte in Geschäftsprozesse.

[GAR14a] fasst die genannten Herausforderungen in vier technischen Aspekten eines *Enterprise Mobility Managements* zusammen. Erstens ist zu eruieren, wie Smart-Mobilgeräte geeignet in die *Enterprise Architecture* integriert werden können. Zweitens muss geklärt werden, wie mit der Schnelllebigkeit (*Technology Evolution*) von Smart-Mobilgeräten adäquat umgegangen werden soll - sowohl die Hardware - als auch die Softwarezyklen sind immer kürzer und müssen angemessen berücksichtigt werden. Drittens muss verstanden werden, welche Aspekte (z.B.

¹Gartner Inc. [GAR14b]

²*bring your own device*

Prozesse, Anwenderfähigkeiten, Nutzerakzeptanz und Werkzeuge) im Zusammenspiel betrachtet werden müssen, um mobile Anwendungen (*Mobile Applications*) erfolgreich zu entwickeln und in Betrieb zu nehmen. Schließlich muss das Thema der Sicherheit sorgfältig einbezogen werden.

Insbesondere die dritte Fragestellung ist im Kontext mobiler Aktivitäten wichtig. Werden zum Beispiel existierende Prozesse im Unternehmen oder der Aspekt der Nutzerakzeptanz vernachlässigt, sind die entwickelten mobilen Anwendungen meist zum Scheitern verurteilt [GAR14c].

1.2 Beitrag

In dieser Arbeit wird im Kontext des *Enterprise Mobility Managements* behandelt, wie Smart-Mobilgeräte in *PrMT* integriert werden können. Dieses Ziel wird im Folgenden weiter verfeinert. Generell wird *PrMT* dazu verwendet, die Prozesse eines Unternehmens digital abzubilden und zu automatisieren. Fachliche Prozesse werden ganz oder teilweise in digitale Prozesse überführt, die automatisiert vom Prozess-Management-System (*PrMS*) ausgeführt werden. Daneben haben sich *PrMS* als Integrationsinstrument für IT-Systeme in Unternehmen etabliert. Ein *PrMS* ruft dazu bei der Ausführung des digitalen Prozesses die spezifizierten Anwendungsdienste (z.B. Datenbankanwendungen) auf und ermöglicht so einen flexiblen Datenaustausch zwischen ihnen. Im Allgemeinen umfassen die Schritte des digitalen Prozesses, die durch sog. Aktivitäten repräsentiert werden, nicht nur Aufrufe von IT-Systemen, sondern auch menschliche Interaktionen. Im medizinischen Umfeld könnte eine solche Interaktion bedeuten, dass ein Krankenhausarzt für einen Patienten ein Röntgenbild benötigt und er dazu eine Röntgenuntersuchung in der Radiologieabteilung anfordert. Aktivitäten, die solche menschlichen Interaktionen im digitalen Prozess abbilden, werden als *interaktive Aktivitäten* bezeichnet. Dagegen werden Prozessschritte, die automatisiert Aufrufe von IT-Systemen im digitalen Prozess abbilden, als automatische Aktivitäten bezeichnet. Bisher werden interaktive Aktivitäten hauptsächlich für *PrMS* mit stationären Desktop-Systemen³ betrachtet. Der Arzt fordert zum Beispiel mit einer Anwendung auf seinem Desktop-System die Röntgenuntersuchung in der Radiologie an. Die Anwendung kommuniziert dann mit dem *PrMS*, um die eingegebenen Daten mit der Radiologie auszutauschen.

Entlang des *Enterprise Mobility Management* Trends ergibt sich nun zusätzlich die Anforderung, interaktive Aktivitäten bei Bedarf auch auf Smart-Mobilgeräten auszuführen. So könnte zum Beispiel ein Arzt fordern, dass er direkt am Patientenbett mit seinem Smart-Mobilgerät die Röntgenuntersuchung anfordern kann. Bisher muss er dazu in sein Arztzimmer gehen, um diese Leistungsanforderung über das entsprechende Desktop-System zu erfassen. Dies ist mit einem gewissen Zeitaufwand verbunden; außerdem kann es infolge von Zwischenfällen anderer Patienten dazu kommen, dass die Röntgenuntersuchung erst mit großer Zeitverzögerung angeordnet wird, da der Arzt beim Gang in sein Zimmer durch die Zwischenfälle unterbrochen wird. Bisher existieren nur wenige Ansätze, die interaktive Aktivitäten eines *PrMS* auf Smart-Mobilgeräten ermöglichen. Das Ziel dieser Arbeit ist es, die interaktive Ausführung von Aktivitäten auf Smart-Mobilgeräten robust zu unterstützen—wir sprechen in diesem Kontext von interaktiven mobilen Aktivitäten. Im Zusammenhang mit [GAR14a] wurde bereits diskutiert, dass für das *Enterprise Mobility Management* verschiedene Aspekte im Zusammenspiel betrachtet werden müssen. Diese Arbeit befasst sich mit der Ausführung mehrerer interaktiver mobiler

³Im Folgenden als Desktop-Systeme bezeichnet.

Aktivitäten mit einem Fokus auf Robustheit. Die robuste Ausführung interaktiver mobiler Aktivitäten ist vor dem Hintergrund der Akzeptanz durch Endanwender entscheidend. Studien haben gezeigt, dass speziell in mobilen Umgebungen die kognitive Belastung für Endanwender hoch ist [Ost07] und sie daher vom System bei ihren Aufgaben möglichst optimal unterstützt werden sollten.

Wir illustrieren die wichtigsten Beiträge der Arbeit entlang des Beispiels aus Abbildung 1.1. Es zeigt den Prozess zur Durchführung einer Röntgenuntersuchung im Krankenhaus. Es sind ausschließlich interaktive Aktivitäten dargestellt, die entweder von einer Pflegekraft (vgl. Abb. 1.1①), einem Arzt (vgl. Abb. 1.1②) oder der Radiologie (vgl. Abb. 1.1③) durchgeführt werden. Angemerkt sei ferner, dass jede interaktive Aktivität des Prozesses von einem anderen Endanwender ausgeführt werden kann. Die interaktiven Aktivitäten des Arztes drücken lediglich aus, dass diese von einem Endanwender mit der Rolle Arzt durchgeführt werden müssen.

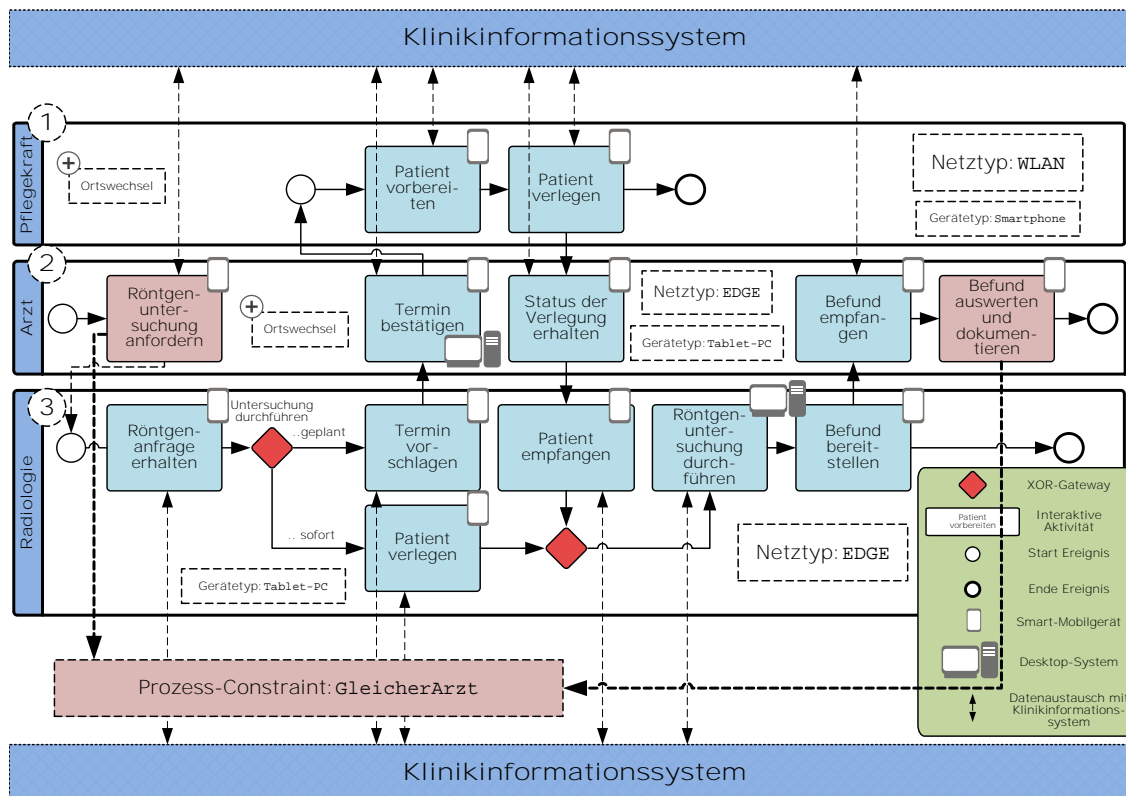


Abbildung 1.1: Klinikprozess einer Röntgenuntersuchung in BPMN-Notation

Der **erste Beitrag** dieser Arbeit betrifft die Ausführung interaktiver mobiler Aktivitäten. Um diese zu ermöglichen müssen vier Fragestellungen adressiert werden. Erstens muss der mobile Kontext, der bislang für interaktive Aktivitäten nicht ausreichend betrachtet wurde, geeignet erfasst werden. Zweitens muss mithilfe dieses Kontexts ermittelt werden können, welche mobilen Endanwender tatsächlich geeignet sind, eine bestimmte interaktive mobile Aktivität auszuführen. Wurde ein konkreter mobiler Endanwender für die Ausführung der Aktivität gefunden, muss drittens der mobile Kontext auch bei der Ausführung der Aktivität berücksichtigt werden. Viertens muss sichergestellt werden, dass sowohl die Zuteilung geeigneter mobiler Endanwender zu einer interaktiven mobilen Aktivität als auch deren Ausführungsunterstützung

in automatischer Art und Weise durchgeführt wird. Aufgrund der komplexen Umgebung mit ihren vielen Parametern sollten manuelle Entscheidungen vermieden werden. Letzteres ist insbesondere notwendig, um die Robustheit bei der Ausführung interaktiver mobiler Aktivitäten zu steigern.

Im Beispielprozess aus Abbildung 1.1 arbeitet die Pflegekraft über ein WLAN mit dem Smartphone, der Arzt hingegen arbeitet über eine EDGE-Verbindung mit einem Tablet-PC. Sowohl bei der Pflegekraft als auch dem Arzt ist vermerkt, dass für sie häufige Ortswechsel zu erwarten sind, was für die Endanwender der Radiologie wiederum nicht zutrifft. Entsprechende Parameter der mobilen Umgebung müssen geeignet erfasst werden, um einerseits den geeignetsten mobilen Endanwender für die Ausführung zu bestimmen (z.B. Endanwender mit großer Nähe zum Ausführungsort) und andererseits eine Unterstützung des mobilen Endanwenders während der Ausführung zu ermöglichen, etwa wenn er das Netz verliert.

Der **zweite Beitrag** der Arbeit ist die Konzeption einer geeigneten Ausnahmebehandlung für interaktive mobile Aktivitäten. Eine solche Ausnahmebehandlung ist unabdingbar, um die angestrebte Robustheit realisieren zu können. Im Vergleich zu Desktop-Systemen können Smart-Mobilgeräte die Verbindung verlieren. Das Smart-Mobilgerät kann zudem jederzeit crashen bzw. infolge einer niedrigen Akkuleistung ausfallen. Um solchen Situationen robust zu begegnen, muss eine geeignete Ausnahmebehandlung etabliert werden. Diese wiederum muss in automatischer Art und Weise realisiert und bewerkstelligt werden, ebenfalls wieder mit dem Ziel einer gesteigerten Robustheit.

Schließlich muss der mobile Kontext berücksichtigt werden, um eine geeignete Ausnahmebehandlung für Endanwender zu bieten. Im Prozess aus Abbildung 1.1 wird zum Beispiel für die mobile Aktivität *Patient vorbereiten* eine Ausnahmebehandlung notwendig, wenn das Smart-Mobilgerät der ausführenden Pflegekraft infolge eines niedrigen Akkustand ausfällt.

Der **dritte Beitrag** der Arbeit ist die angemessene Berücksichtigung der Prozessumgebung bei der Ausführung interaktiver mobiler Aktivitäten. Existierende Ansätze behandeln Eigenschaften der Prozessumgebung kaum, obwohl sie für die Praxis relevant sind. Folgende drei Aspekte sollen in diesem Zusammenhang behandelt werden:

- Prozess-Constraints sollen auch für Prozesse mit mobilen Aktivitäten berücksichtigt werden. In Abbildung 1.1 etwa gibt es ein Prozess-Constraint zwischen den Aktivitäten *Röntgenuntersuchung anfordern* und *Befund auswerten und dokumentieren*. Dieser besagt, dass beide Aktivitäten vom gleichen Arzt ausgeführt werden sollen. Solche Prozess-Constraints sind im Zusammenspiel mit anderen Konzepten, wie dem mobilen Kontext, der mobilen Ausführungsunterstützung oder der skizzierten Ausnahmebehandlung in ihrer Realisierung nicht trivial.
- Die Unterstützung interaktiver mobiler Aktivitäten kann sich auch auf interaktive Aktivitäten, die auf Desktop-Systemen ausgeführt werden, auswirken. Beispielsweise sollte ein mobiler Endanwender von einer mobilen Ausführung zu einer Ausführung auf dem Desktop-System wechseln können. Bezogen auf Abbildung 1.1 etwa könnte die Aktivität *Befund auswerten und dokumentieren* auf einem Smart-Mobilgerät begonnen, dann aber auf einem Desktop-System beendet werden.
- Es werden Architekturaspekte berücksichtigt, um die Ausführung interaktiver mobiler Aktivitäten in bestehende *PrMS* integrieren zu können. So muss analysiert werden, ob existierende Anwendungen für Desktop-Systeme zur Ausführung interaktiver Aktivitäten ohne Änderungen auf Smart-Mobilgeräte übertragen werden können.

Insgesamt entwickelt diese Arbeit fortschrittliche Konzepte zur robusten Ausführung interaktiver mobiler Prozess-Aktivitäten. Diese Konzepte werden darüber hinaus in einem Rahmenwerk, das aufzeigt, wie sie im Zusammenspiel realisiert werden können, integriert. Das Rahmenwerk wird prototypisch realisiert und der resultierende Prototyp praktisch evaluiert, um die praktische Anwendbarkeit der Konzepte zu demonstrieren.

1.3 Methodik

Generell kommen in der Forschung zu Informationssystemen drei Methoden zum Einsatz: *Design Science*, *Natural Science* [MS95] und *Behavioral Science* [HMPR04]. Auf eine detaillierte Diskussion wird an dieser Stelle verzichtet (vgl. [Wie14]). Für die vorliegende Arbeit wird der *Design Science*-Ansatz zugrunde gelegt. Da sich dieser an einer existierenden Technologie orientiert [HMPR04], die für diese Arbeit in Form der Prozessumgebung gegeben ist, ist seine Anwendung naheliegend. Um *Design Science* umzusetzen, sind mehrere Aspekte zu berücksichtigen. Diese werden in [HMPR04] in Form von sieben Leitlinien zusammengefasst, die auch im Rahmen dieser Arbeit ihre Anwendung finden. Ferner erfolgt eine spezifische Diskussion der jeweils angewandten Methodik in einzelnen Kapiteln dieser Arbeit. Tabelle 1.1 zeigt die sieben Leitlinien des *Design Science*-Ansatzes und deren Umsetzung im Rahmen dieser Arbeit.

1.4 Aufbau

Abbildung 1.2 fasst den Aufbau der Arbeit, die aus insgesamt fünf Teilen besteht, zusammen. Teil I beschreibt das adressierte Problem und führt eine Anforderungsanalyse durch. Teil II wiederum führt die Prozessumgebung ein, die in dieser Arbeit zugrunde gelegt wird. Teil III stellt das entwickelte Rahmenwerk zur robusten Ausführung interaktiver mobiler Aktivitäten vor, während Teil IV sich mit der Validation des Rahmenwerks befasst. In Teil V schließt die Arbeit mit einer Zusammenfassung und einem Ausblick.

	Leitlinie	Umsetzung
L1	Design des Artefakts	Die Anforderungen zur robusten Ausführung mobiler Aktivitäten werden aufgestellt und daraus das Rahmenwerks abgeleitet. Letzteres stellt das Artefakt des <i>Design Science</i> -Ansatzes dar. Es setzt u.a. die drei genannten Beiträge aus Abschnitt 1.2 um.
L2	Problemrelevanz	Die Problemrelevanz wird durch mehrere Fallstudien und eine Literaturrecherche nachgewiesen.
L3	Design Bewertung	Die Fallstudien und die Literaturrecherche werden in mehreren Zyklen durchgeführt, um das Artefakt iterativ zu verfeinern.
L4	Forschungsbeitrag	Durch eine intensive Betrachtung des Stands der Technik konnte der innovative Forschungsbeitrag des Artefakts nachgewiesen werden.
L5	Forschungsschlüssigkeit	Die Forschungsschlüssigkeit ergibt sich einerseits aus der Anwendbarkeit des Artefakts, andererseits aus den Methoden, die für dessen Design angewandt werden. Durch prototypische Realisierung des Artefakts sowie seiner praktischer Anwendung, wird dessen Anwendbarkeit nachgewiesen. Ferner wird zum Design des Artefakts für jeden Teilaspekt ausgeführt, wie dieser methodisch hergeleitet wird.
L6	Design als Suchprozess	Gemäß <i>Design Science</i> -Ansatz sollen bei der Suche eines geeigneten Designs für das Artefakt die Gesetze der Umgebung beachtet werden. Dieses Vorgehen wird in der vorliegenden Arbeit durch die explizite Berücksichtigung der Prozessumgebung, in der interaktive mobile Aktivitäten ausgeführt werden, umgesetzt.
L7	Kommunikation des Forschungsbeitrags	Das Artefakt soll sowohl Fachexperten als auch Endanwendern bekannt gemacht werden. Das Artefakt wird auf mehreren wissenschaftlichen Konferenzen diskutiert und durch Publikationen den Fachexperten vorgestellt. In diesem Zusammenhang wird das Artefakt auch in praktischen Demos mit den Fachexperten diskutiert. Bei seiner praktischen Anwendung wird mit Anwendern interagiert, um die Anwendbarkeit des Artefakts zu ermitteln.

Tabelle 1.1: Leitlinien des *Design Science*-Ansatzes und deren Umsetzung

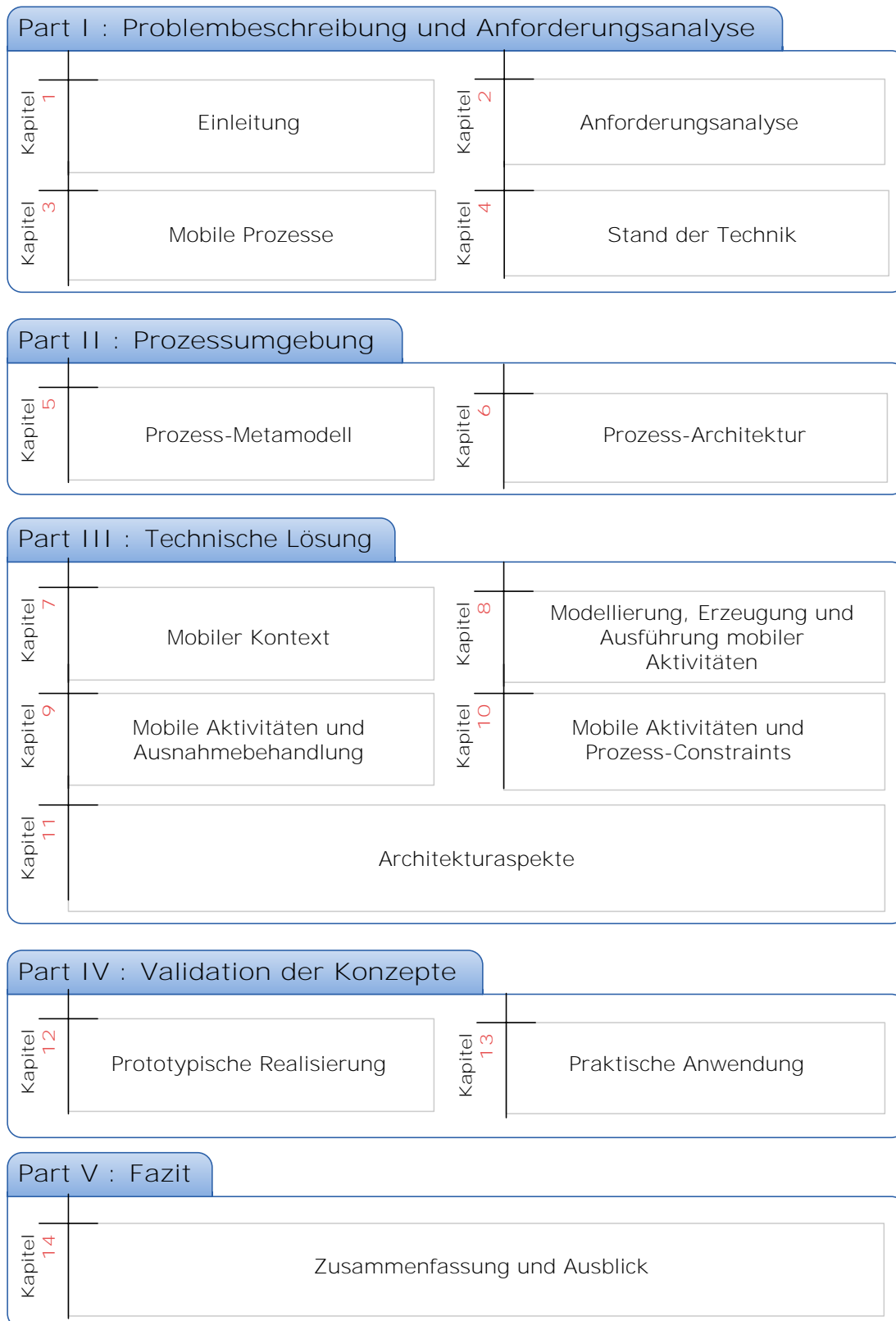


Abbildung 1.2: Aufbau der Arbeit

Jeder dumme Junge kann einen Käfer zertreten. Aber alle Professoren der Welt können keinen herstellen.

Arthur Schopenhauer (1788-1860)

2

Anforderungsanalyse

Die Unterstützung interaktiver mobiler Aktivitäten¹ stellt eine komplexe Aufgabe dar. Einerseits sind technische Herausforderungen (z.B. beschränkte Ressourcen oder Diskonnektivität) angemessen zu berücksichtigen, andererseits zeigt der aktuelle Trend von Smart-Mobilgeräten, dass die Akzeptanz für eine mobile Unterstützungslösung seitens Anwendern einen wesentlichen Aspekt darstellt. Vor allem sollte genau analysiert werden, welche Funktionen unterstützt werden sollen und welche Randbedingungen es zu beachten gilt. Obwohl erhobene Anforderungen darauf abzielen sollten, einen möglichst generischen Ansatz zu entwickeln, dürfen dessen Grenzen nicht außer Acht gelassen werden. Dass wir uns in einem komplexen Forschungsfeld bewegen, wird durch eine Studie der Universität von Massachusetts [Ost07] deutlich. In dieser werden ubiquitäre Prozesse [HCS05], als Obermenge der mobilen Prozesse, und deren Charakteristika in verschiedenen Anwendungsdomänen betrachtet. Die Studie zeigt, dass die Unterstützung mobiler Prozesse und Aktivitäten entscheidend dazu beitragen kann, die Arbeitseffizienz zu steigern. Allerdings müssen deren Charakteristika umfassend analysiert werden. Um die Anforderungen an das in dieser Arbeit entwickelte Rahmenwerk zu erheben, wurde einerseits eine umfangreiche Literaturrecherche durchgeführt, andererseits wurden mehrere Fallstudien durchgeführt.

Kapitel 2 gliedert sich wie folgt: Abschnitt 2.1 diskutiert die Literaturrecherche, während Abschnitt 2.2 auf die durchgeführten Fallstudien eingeht. In Abschnitt 2.3 werden die letztlich identifizierten Anforderungen diskutiert. Abschnitt 2.4 fasst das Kapitel zusammen.

2.1 Literaturrecherche

Die durchgeführte Literaturrecherche gliedert sich in zwei Phasen:

1. In der ersten Phase (*Literaturrecherephase Suchmaschinen*) wurde zum Forschungsgebiet mobiler Prozesse (engl. *Mobile Processes*) eine umfassend Literaturstudie durchgeführt. Dazu wurden folgende Quellen verwendet: Google [Goo14], Google Scholar [GSc14], Microsoft Academic Search [MAS14] und Scopus [Sco14]. Bei der Durchführung der Litera-

¹Im Folgenden wird der Einfachheit halber von mobilen Aktivitäten gesprochen.

turstudie wurde in allen Quellen der Suchbegriff *Mobile Process* zuzüglich Abwandlungen (z.B. Mobile Workflow) verwendet.

2. In der zweiten Phase (*Literaturrecherephase Clusteranalyse*) wurde mit der auf Literaturstudien spezialisierten Software BibTechMon [NFS02] gearbeitet. Diese Software analysiert eine Menge von Publikationen auf Forschungscluster. Letztere kann die Software vielfältig bilden. So könnte die Software bezogen auf eine Menge von Publikationen das Cluster *NoSQL* [NoS14] als Trend im Datenbankbereich identifizieren. Ebenso könnte festgestellt werden, wenn bezogen auf eine gegebene Menge an Publikationen ein bestimmter Forscher heraussticht. Ein weiteres Ergebnis könnte schließlich sein, dass zwei Autoren eng zusammenarbeiten.

Die Analyse mit BibTechMon untergliedert sich in zwei Sub-Phasen. In der ersten Sub-Phase wurde auf Google Scholar der Suchbegriff *Mobile Process* sowie Abwandlungen davon angewandt. Daraus resultierten 61.100 Treffer, die anschließend mit BibTechMon analysiert wurden. Dabei wurde der Suchbegriff erweitert, um das Forschungsgebiet des *Service Oriented Computing* [BBP09, Buc12, BBTR14, BBR11, BBR13], das einen großen Einfluss auf *Mobile Processes* hat, angemessen zu berücksichtigen. Daher wurde in einer zweiten Phase der Suchbegriff *Mobile Process* mit dem Suchbegriff *Service Oriented Computing* zu *Mobile Process and Service Oriented Computing* kombiniert (wiederum zuzüglich Abwandlungen). Dies führte zu weiteren 113.000 Treffern, die mit BibTechMon analysiert wurden.

Die beiden Phasen werden nachfolgend detailliert.

2.1.1 Literaturrecherephase 1: Suchmaschinen

In der ersten Phase wurden mittels der genannten vier Quellen folgende Suchbegriffe ausgewertet (vgl. Tabelle 2.1):

Mobile Activity	Mobile Process	Mobile Workflow	Mobile Information System
Mobile Agent	Distributed Mobile Workflow	Distributed Mobile Process	Mobile Architecture
Mobile Middleware		Ubiquitous Process	
Mobile Application Development and Apps		Smart Mobile Apps and Mobile Business Apps	

Tabelle 2.1: Suchbegriffmenge der Literaturrecherche

Bei der Analyse mit BibTechMon zeigte sich, dass die folgenden Suchbegriffe zusätzlich einbezogen werden sollten (vgl. Tabelle 2.2):

Service Oriented Computing	Agile Systems	Mobile Commerce	Industry 4.0 [i4014]
Mobile Healthcare		Mobile E-Services	

Tabelle 2.2: Erweiterte Suchbegriffmenge der Literaturrecherche

Die Recherche hat schließlich aus der Gesamttrefferzahl zu **24** Arbeiten geführt (vgl. Tabelle 2.3), die zur Anforderungsanalyse relevant waren. Obwohl sich mit BibTechMon eine größere

Anzahl an Ergebnisse analysieren ließ, kann mithilfe der Software auch die Menge besonders relevanter Ansätzen identifiziert werden. Dazu wird die Cluster-Analyse angewandt.

2.1.2 Literaturrecherchephase 2: Clusteranalyse

In der zweiten Phase wurde die Software BibTechMon eingesetzt. Dazu wurde die Phase in zwei Sub-Phasen unterteilt. In der ersten Sub-Phase wurde mittels der 61.100 Google Scholar Suchergebnisse zum Begriff *Mobile Process* (s. Abschnitt 2.1.1) eine Clusteranalyse durchgeführt. Das Ergebnis zeigt Abbildung 2.1. Aus den Clustern wiederum ermittelt BibTechMon, welche Arbeiten am relevantesten innerhalb des jeweiligen Clusters sind. Die Ergebnisse dazu können in [Woh12] nachgelesen werden. Generell wird die Relevanz anhand von Kriterien, wie Menge der Zitate, zeitlicher Verlauf der Zitate und Zusammenarbeit zwischen Forschern ermittelt. **16** der in den Forschungsclustern relevantesten Ansätze (vgl. Tabelle 2.3) haben zur Anforderungsanalyse, zusätzlich zu den **24** in Phase I identifizierten Arbeiten beigetragen.

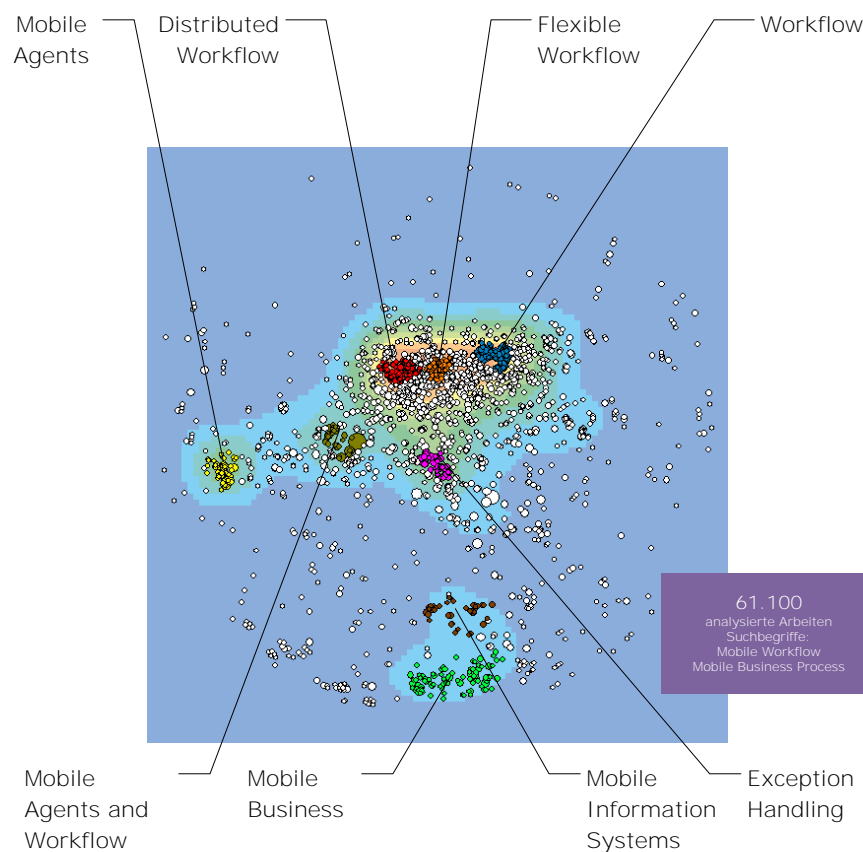


Abbildung 2.1: BibTechMon-Analyse Sub-Phase 1

In der zweiten Sub-Phase wurden die 113.000 Google Scholar Ergebnisse zum Suchbegriff *Mobile Process and Service Oriented Computing* analysiert. Die identifizierten Cluster sind in Abbildung 2.2 zu sehen. Innerhalb dieser Cluster konnten weitere **9** relevante Arbeiten (vgl. Tabelle 2.3) für die Anforderungsanalyse identifiziert werden.

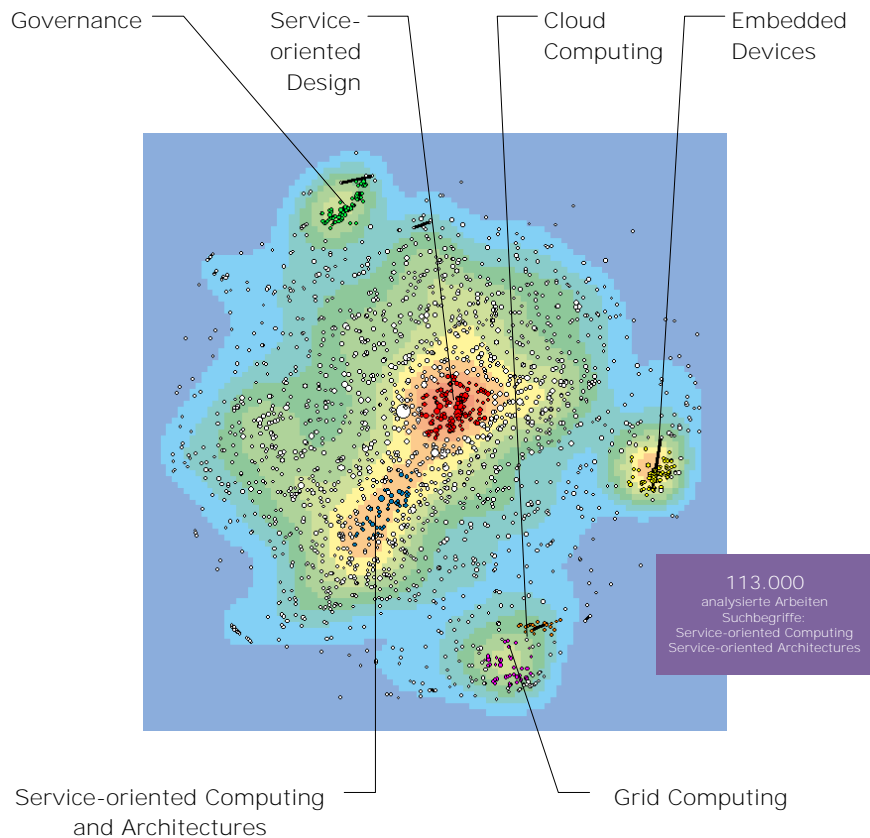


Abbildung 2.2: BibTechMon-Analyse Sub-Phase 2

2.1.3 Ergebnis der Literaturrecherche

Insgesamt wurden aus den beiden Phasen zur Literaturrecherche **49** relevante Arbeiten für die Anforderungsanalyse identifiziert. Diese Arbeiten werden in Tabelle 2.3 zusammengefasst.

Phase	Identifizierte Arbeiten der Literaturrecherche
Recherche mittels Suchmaschinen	[ZHKL10, HS09, RSJ07, SGDD07, UELW10, LCS05, CK00, CGR ⁺ 13, Mue04, AGA ⁺ 95, Sat96, MM05, BD00a, KZL06, HHGR06, ZDL09, KS92, PB99, DPS ⁺ 94, DLNW13, SGS ⁺ 02, BMR96, RD97, HST07]
BibTechMon-Analyse Sub-Phase 1	[SW03, PS93, WSNL07, SH06, PMLR14, KZL06, PTKR10, KG04b, ABBH00, RS03, BBG06, Puc01, SARA04, LKHK05, Kir03, MGO ⁺ 13]
BibTechMon-Analyse Sub-Phase 2	[SF03, Ost07, SKH07, Sch09, SKHR08, FGL ⁺ 96, GGG099, KL10, Mar09]

Tabelle 2.3: Literaturrecherche mittels Suchmaschinen und BibTechMon-Analyse

Diese Arbeiten stellen die Basis der Literaturrecherche dar, um Anforderungen an eine robuste und kontextbezogene mobile Aktivitätenausführung (s. Abschnitt 2.3) zu formulieren. Ferner werden die Analyse-Ergebnisse als Basis für die im Folgenden durchgeführten Fallstudien herangezogen.

2.2 Fallstudien

Dieser Abschnitt fasst die zwecks Anforderungserhebung durchgeführten Fallstudien zusammen. Konkret wurden Fallstudien in drei Anwendungsdomänen durchgeführt (vgl. Tabelle 2.4). Bevor diese im Detail vorgestellt werden, sollen wichtige Hintergrundinformationen gegeben werden:

- Ein wichtiges Kriterium bei der Auswahl der Fallstudien war, dass eine Prozessunterstützung bereits vorhanden war. Kam in der jeweils betrachteten Umgebung bislang kein *PrMS* zum Einsatz, wurden neue Prozesse² erhoben. Die Fallstudien, zu denen neue Prozesse erhoben werden mussten, sind in Tabelle 2.4 in der letzten Spalte markiert.
- Auch wenn teilweise bereits Prozessmodelle vorlagen, wurde für jede Fallstudie analysiert, ob weitere Prozesse gefunden werden können. Falls ja wurden diese mittels BPMN [BPM14a] modelliert und ebenfalls für die Anforderungserhebung herangezogen (vgl. Tabelle 2.4).
- In einer Fallstudie zur Patientenvisite wurden vier Abteilungen besucht, was für die Anforderungserhebung hilfreich war. Insbesondere konnten aussagekräftige Unterschiede bzgl. der Anforderungen an eine robuste und kontextbezogene Ausführung mobiler Aktivitäten identifiziert werden.
- Zu den Fallstudien wurden mobile Anwendungen prototypisch entwickelt (vgl. Tabelle 2.4), die Einblicke in die Unterstützung mobiler Aktivitäten in einer Prozessumgebung ermöglichen sollten. Ein wichtiger Aspekt bei der Entwicklung der mobilen Anwendungen war es, jeweils eine geeignete Informationsarchitektur zu entwerfen [IAT14].
- Es wurde besonderes Augenmerk darauf gelegt, was mobile Endanwender im Ausnahmefall (z.B. Netzaabbrüche) als Gegenmaßnahme wünschen.

Abbildung 2.3 zeigt die bei allen Fallstudien angewandte Methodik.

2.2.1 Klinische Visite

Die erste Fallstudie bezieht sich auf Klinische Visiten. Generell sind Abläufe in Kliniken besonders interessant hinsichtlich einer mobilen Unterstützung [BTR10]. Abbildung 2.4 zeigt die Klinischen Abteilungen, in denen im Kontext von Fallstudien mobile Aktivitäten untersucht wurden.³

Generell wurden die Visiten nach dem gleichen Schema analysiert. Konkret wurden folgende Phasen durchlaufen:

²Auch eine Papierform zur Erhebung von Prozessen war zweckdienlich.

³Weitere Informationen zu den verwendeten Kliniksystemen SAP ERP und MCC Meierhofer können in [SAP14c, MCC14] nachgelesen werden.

Anforderungen	Bemerkungen Szenario	Prozess- erhebung notwendig
Krankenhaus Visite Prototyp:Ja Referenzen: [Lan12, Eic12, PLRH12, PMLR14, Gei14]	Es wurden in drei klinischen Abteilungen Prozesse zur Visite erhoben. Für jeden Prozess wurde ermittelt, welche Prozessaktivitäten aus Fachsicht mobil ausgeführt werden sollen. Ferner wurde detailliert analysiert, welche Daten mobil zur Verfügung stehen sollen. Besondere Merkmale des Szenarios: Hohe Dynamik, viele Ausnahmen, potentiell viele Akteure, umfangreiche Kontextinformationen, viele Sensormessungen.	✓
Luftfahrt Catering Prototyp:Ja Referenzen: [Sto11, Sto10]	Es wurde der Prozess zur Essensbelieferung von Flugzeugen analysiert. Dieser reicht von der Zubereitung der Mahlzeiten bis zur Einlagerung des Essens im Flugzeug. Schlussendlich müssen leere Essensbehälter zurücktransportiert werden. Für diesen Prozess wurde ebenfalls analysiert, für welche Aktivitäten eine mobile Ausführung infrage kommt. Besondere Merkmale des Szenarios: Wenig dynamisch, viele wechselnde Netze und Abbrüche, viele Akteure, langlaufende Aktivitäten, große Laufwege, viele Sensormessungen.	✓
Telecare Prototyp:Ja Referenzen: [Kit12, Jab13, Ene11]	Mit Hausärzten wurde analysiert, wie deren Prozesse zwischen der Behandlung von Patienten in der Praxis und deren Versorgung bei Hausbesuchen mobil unterstützt werden kann. Besondere Merkmale des Szenarios: Dynamisch, potentiell viele wechselnde Netze und Abbrüche, sehr wenig Akteure, viele Sensormessungen, intensive Dateneingabe mit wechselndem Kontext, Offline-Anforderungen.	✓

Tabelle 2.4: Fallstudien

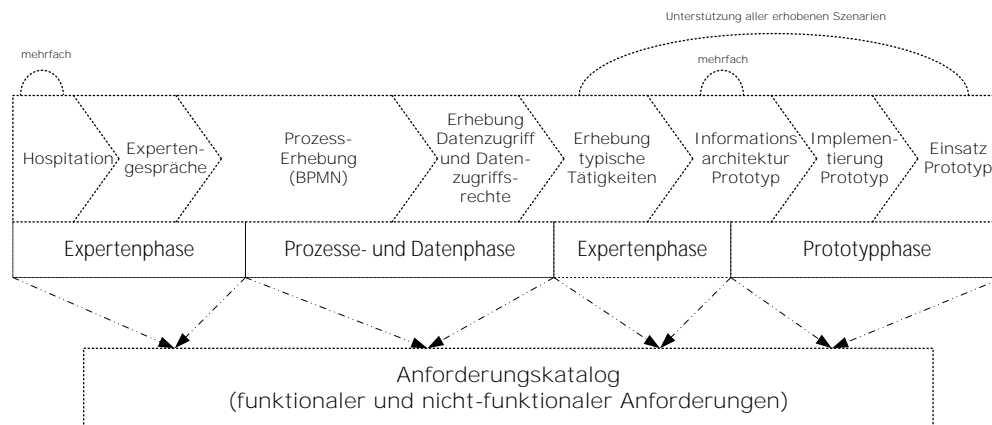


Abbildung 2.3: Angewandte Methodik

	Innere Medizin	Notfallaufnahme	Orthopädie	Unfallchirurgie
Besuchte Klinik	Uniklinik Ulm	Uniklinik Ulm	RKU Ulm	Uniklinik Ulm
Anzahl Betten	25	12	35	>100
Durchschnittliche Aufenthaltsdauer der Patienten	Tage, Wochen	Stunden	Wochen, Monate	1-30 Tage
Anzahl Visiten	2x / Tag	3x / Tag	1x / Tag	1x / Woche
Visitegrund	Täglichen Überblick gewinnen	Informationsaustausch zwischen Schichtwechseln	Täglichen Überblick gewinnen	Überblick für Chefarzt
Involvierte Personen	Zwei Stationsärzte, zwei Schwestern oder Pfleger	6-8 Stationsärzte, 1 Oberarzt, evtl. Physiotherapeuten oder Konsiliarärzte	3 Stationsärzte, 1-3 Schwestern Verschiedene Therapeuten	Chefarzt, Leitender Oberarzt, Weitere Stationsärzte
Behandlungszeit pro Patient	7 + 3.5 Minuten (Vorbereitung + Behandlung)	4 Minuten	4 Minuten	3 Minuten
Klinisches Informationssystem	SAP, ERP Software	SAP, ERP Software	MCC, Meierhofer AG	SAP, ERP Software
Mobil verfügbare Daten				
Vitaldaten	✓ (Schwestern-Mitschrieb)	✓ (Schwestern-Mitschrieb)	✓ (Schwestern-Mitschrieb)	✓ (Schwestern-Mitschrieb)
Medikation	✓ (Schwestern-Mitschrieb)	✓ (Schwestern-Mitschrieb)	✓ (Schwestern-Mitschrieb)	✓ (Schwestern-Mitschrieb)
Röntgenbilder	✗	✗	✓ (Tablet PC)	✓ (Ausdruck)
Laborbefunde	✗	✗	✓ (Tablet PC)	✓ (Ausdruck)
(✓) : verfügbar (✗) : nicht verfügbar				

Abbildung 2.4: Beteiligte Klinische Abteilungen in Fallstudie 1

1. Jede der untersuchten Visiten (vgl. Abb. 2.4) wurde dreimal begleitet, sodass eine umfangreiche Dokumentation der Arbeitsschritte resultierte. Letztere wurden in drei Phasen unterteilt (vgl. Abb. 2.4), d.h. die Phase vor dem Patientenraum, die Phase im Patientenraum und die Phase der nachgelagerten Dokumentation (vgl. [PMLR14]).
2. Im Anschluss wurden BPMN-Prozessmodelle zu den erfassten Abläufen erstellt. Abbildung 2.5 zeigt ein Beispiel eines solchen Prozessmodells.
3. Die Prozessmodelle wurden anschließend mit den Fachexperten in den Kliniken diskutiert. Hier zeigte sich, dass die erfassten Prozessmodelle gut von den Klinikern verstanden wurden.
4. Auf Basis der Prozessmodelle wurde für jeden Visitetyp analysiert, an welchen Stellen eine mobile Unterstützung hilfreich wäre. Ferner wurde analysiert, was sich Kliniker für die mobile Unterstützung wünschen (Anforderungen) und welche besonderen Herausforderungen sie sehen (vgl. Markierungen in Abb. 2.5). Erwähnenswert ist, dass sich bei einer der untersuchten Visiten (vgl. Abb. 2.4) bereits eine einfache mobile Lösung zur Durchführung ausgewählter Tätigkeiten im Einsatz befand. Die Analyse dieses Systems lieferte wertvolle Einblicke.
5. In einem letzten Schritt wurde ein mobiler Prototyp entwickelt, mit dem eine Auswahl von Tätigkeiten (vgl. Tabelle 2.5) während der Visite mobil unterstützt durchgeführt wurde. Der Prototyp kam in allen klinischen Abteilungen (vgl. Abb. 2.4) explorativ zum Einsatz.

Zu dieser Fallstudie seien die folgenden Aspekte angemerkt:

1. Während der Anwendung des mobilen Prototyps konnten weitere Prozesse identifiziert und modelliert werden (vgl. Abb. 2.5 für ein Beispiel).
2. Darüber hinaus wurden Datenzugriffe, die mobil möglich sein sollen, identifiziert (vgl. Tabelle 2.6).

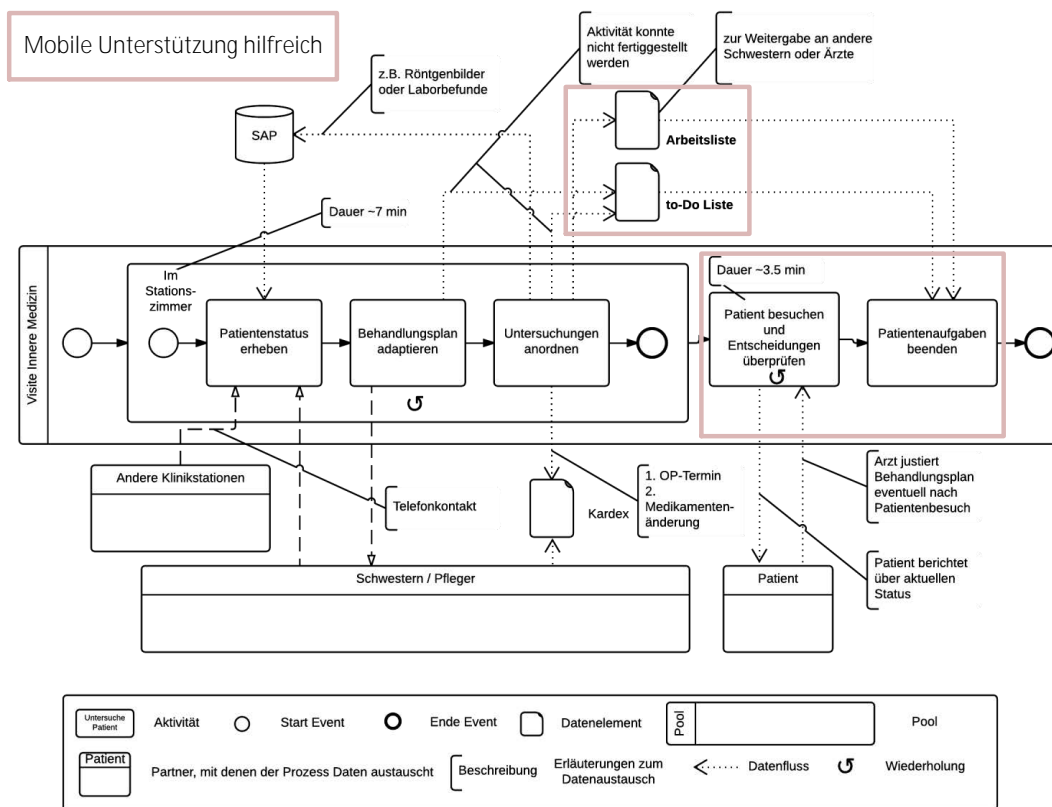


Abbildung 2.5: Visiteprozess der Inneren Medizin

Tätigkeiten	Beschreibung
Tätigkeit I	Ein Visitearzt möchte während seiner Visite spontan Röntgenbilder betrachten. Dazu muss die aktuelle Aufgabe kurzfristig unterbrochen werden können.
Tätigkeit II	Ein Visitearzt möchte während seiner Visite den Status einer angeforderten Röntgenuntersuchung ermitteln.
Tätigkeit III	Ein Visitearzt möchte vor seiner Visite alle Untersuchungsergebnisse der letzten zwei Stunden einsehen.
Tätigkeit IV	Ein Visitearzt möchte vor seiner Visite folgende Informationen für jeden Patienten zusammenstellen: Anamnese, Laborbefunde, Röntgenbilder und Medikationen.
Tätigkeit V	Ein Visitearzt möchte nach der Visite alle anstehenden Aufgaben auf einen Blick sehen.
Tätigkeit VI	Ein Visitearzt möchte während seiner Visite neue Aufgaben anordnen.

Tabelle 2.5: Auswahl typischer Szenarien zur Klinischen Visite

- Es soll die zum behandelten Szenario entwickelte Informationsarchitektur [IAT14] vorgestellt werden (vgl. Abb. 2.7). Sich zu überlegen, wie der Informationsfluss für einen

mobilen Prototypen gestaltet werden muss, ist ein essentieller Faktor für die Akzeptanz des mobilen Systems (vgl. [PMLR14]).

4. Abhängigkeiten zwischen mobilen Endanwendern (z.B. "Zwei Aktivitäten müssen vom selben mobilen Endanwender ausgeführt werden") sind für das Szenario der Klinischen Visite relevant und müssen daher unterstützt werden.
5. Die Ausführung einer mobilen Aktivität auf einem Smart-Mobilgerät zu beginnen und auf einem Desktop-System fortzuführen, wird für viele Tätigkeiten der Klinischen Visite gewünscht.
6. Die angemerkte Behandlung von Ausnahmen (z.B. Diskonnektivität oder Instant-Shutdowns) ist für das Szenario der Klinischen Visite essentiell.

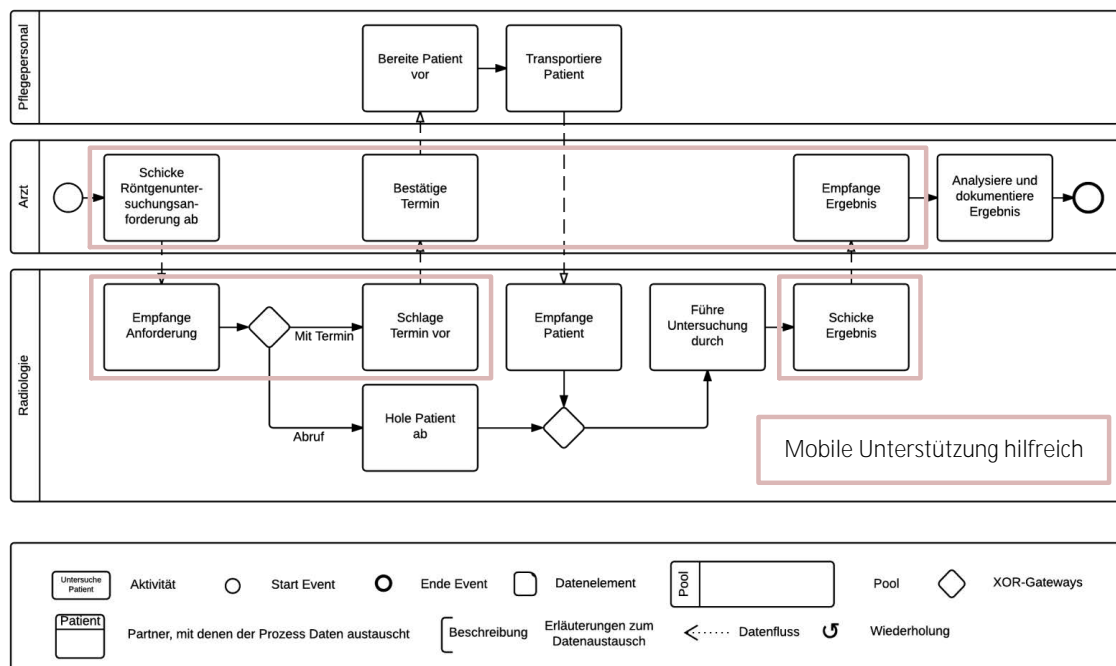


Abbildung 2.6: Prozess der Röntgenuntersuchung

2.2.2 Luftfahrt-Catering

Die zweite Fallstudie wurde im Bereich Luftfahrt- bzw. Airline-Catering durchgeführt. Konkret geht es um den Gesamtprozess zur Beladung eines Flugzeuges mit Essen für den Flug. Dieser Prozess ist komplex und durchläuft mehrere Phasen, die aufeinander abgestimmt sein müssen. Herausfordernd wird der Prozess dadurch, dass die Behälter, die das Essen aufnehmen, in einer globalen Schleife um den Globus wandern. Man stelle sich vor, in Frankfurt werden solche Behälter ein- und in Sydney ausgeladen. Auf deren Wiederbereitstellung in Sydney zu warten, wäre nicht möglich, daher kommen für den Folgeflug andere Behälter zum Einsatz. Die Bewerksstellung dieser Logistik, die sich an jedem Flughafen anders darstellt, ist eine äußerst komplexe Aufgabe. Erschwert wird der Prozess zudem durch die Lebensmittelvorgaben der

Datenart	Zugriffsrecht Arzt		Zugriffsrecht Pflegepersonal	
	lesend	schreibend	lesend	schreibend
Vitaldaten	✓	X	✓	✓
Medikationen	✓	✓	✓	X
Befunde	✓	✓	X	X
Untersuchungstermine	✓	✓	✓	(✓)
Arbeitsplan	✓	✓	✓	(✓)
Diagnostik	✓	X	X	X

✓=benötigt, (✓)=manchmal benötigt, X=nicht benötigt

Tabelle 2.6: Identifizierte Datenzugriffe und Zugriffsrechte

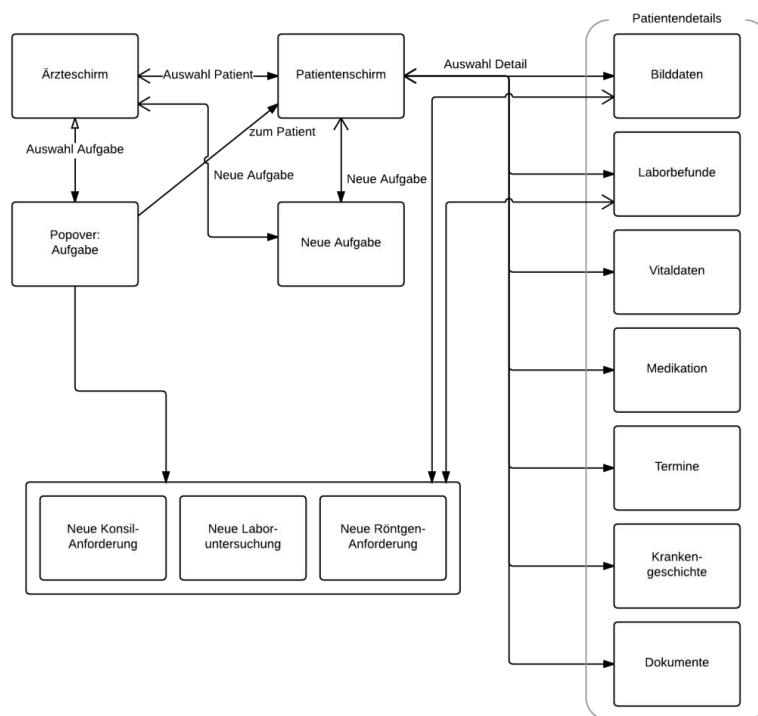


Abbildung 2.7: Informationsarchitektur des mobilen Prototyps der Klinischen Visite

am Flug beteiligten Länder. Zum Beispiel muss länderspezifisch berücksichtigt werden, welche Temperaturen zu welchem Zeitpunkt einzuhalten sind. Insgesamt ist die Gesamtlogistik für das Luftfahrt-Catering komplex.

Der Gesamtprozess lässt sich in den Teil, der außerhalb des Flugzeuges ausgeführt wird (on-ground), und denjenigen, der im Flugzeug ausgeführt wird (on-board) unterteilen. Weiters gliedert sich der Gesamtprozess in insgesamt 10 Phasen, die nachfolgend kurz vorgestellt werden. Für jede Phase wurden die jeweiligen Prozesse in BPMN modelliert.

- **Phase 1 (Wareneingang):** Essen wird beim Caterer angeliefert.

- **Phase 2 (Lager):** Essen wird eingelagert.
- **Phase 3 (Produktionsstation):** Essen wird produziert.
- **Phase 4 (Assemblierung):** Essen wird portioniert.
- **Phase 5 (Kommissionierstation):** Essen wird in Trolleys⁴ gepackt.
- **Phase 6 (Warenausgang und Endkontrolle):** Trolleys werden kontrolliert.
- **Phase 7 (Transport zum Flugzeug):** Trolleys werden zum Flugzeug transportiert.
- **Phase 8 (Inflight):** Trolleywaren werden während des Flugs verarbeitet.
- **Phase 9 (Rücktransport Trolleys):** Trolleys werden am Zielflughafen ausgeladen.
- **Phase 10 (Wartung und Einlagerung):** Trolleys werden wieder einsatzbereit gemacht und eingelagert.

Insgesamt resultierte ein ausführlicher Katalog, der die Herausforderungen des bisherigen Ablaufs gut widerspiegelt. Aktuell ist der Gesamtprozess ohne IT-seitige Unterstützung realisiert. Bei der Analyse der Anforderungen wurden insbesondere wieder Aktivitäten identifiziert, die mobil ausgeführt werden sollen. Bei den Gesprächen mit den Fachexperten hat sich gezeigt, dass die erstellten Prozessmodelle für die Analyse der einzelnen Phasen (Abb. 2.8 zeigt den Prozess zu Phase 5) nützlich waren.

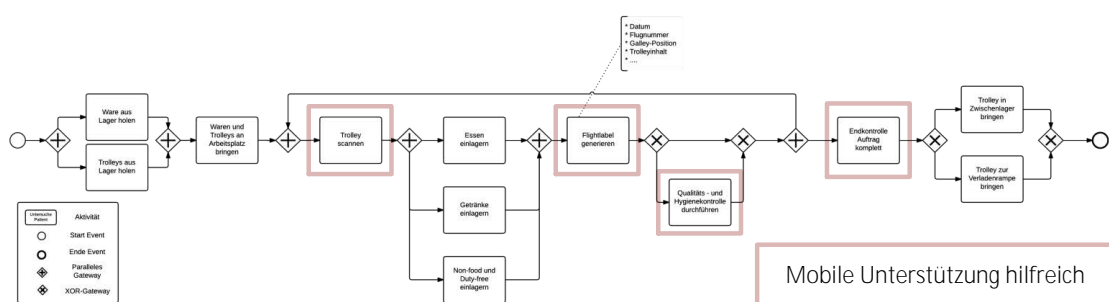


Abbildung 2.8: Prozess zur Phase 5 der Kommissionierung

Abschließend seien zu dieser Fallstudie folgende Aspekte angemerkt:

1. Es ließen sich viele Herausforderungen identifizieren, die sich durch den mobilen Kontext ergeben. Die genannten Phasen laufen allesamt an verschiedenen Orten ab. Dadurch ergeben sich häufige Netzwechsel; an einigen Stellen ist gar kein Netz vorhanden. Weiters müssen Aufgaben oft übergeben werden. Insgesamt stellt sich eine IT-seitige Unterstützung mobiler Aktivitäten sehr komplex dar.
2. Die Behandlung von Ausnahmen (z.B. Diskonnektivität oder Instant-Shutdowns) ist für das Airline-Catering Szenario essentiell.
3. Bisher läuft der gezeigte Catering-Prozess ausschließlich papierbasiert ab, was eine gute Ausgangssituation darstellte, die relevanten Schwachstellen zu identifizieren. Beispielswei-

⁴Trolleys sind die fahrbaren Einheiten, die man aus dem Flugzeug kennt, wenn das Essen serviert wird. Diese sind auf den Mittelgang eines Flugzeugs angepasst.

se ist bei der Kommissionierung (vgl. Abb. 2.8) das manuelle Vergleichen der Flight-Labels mit der Bestellung zeitaufwändig und es führt zudem zu vielen Ablesefehlern.

4. Es kommt vereinzelt bereits ein sensorisches Ablesen von Temperaturdaten zum Einsatz. Weiters sind viele Stellen denkbar, an denen ein sensorisches Erfassen von Werten hilfreich wäre.
5. Es kommen viele IT-Systeme zum Einsatz, die auf Basis der papierbasierten Dokumentation befüllt werden müssen. Es zeigte sich vor allem, dass eine Prozessunterstützung sinnvoll ist. Auch war festzustellen, dass zwischen vielen der Systeme praktisch keine Schnittstellen existieren, was die papierbasierte Informationsübertragung notwendig macht.
6. Das Szenario umfasst viele Personen, zwischen denen Abhängigkeiten abgebildet werden müssen, die teilweise aus unterschiedlichen Organisationen kommen. Auch dies erfolgt bisher ausschließlich auf Papierbasis. Als weiteres Problem sprechen die Beteiligten oftmals unterschiedliche Sprachen, was ihre Kommunikation erschwert und den papierbasierten Prozess noch schwieriger gestaltet.
7. Es muss jederzeit möglich sein, die auf einem Smart-Mobilgerät begonnenen Aufgaben, auf einem Desktop-System fortzuführen.

2.2.3 Telecare

Die dritte Fallstudie befasst sich mit Telecare. Konkret geht es um zwei Szenarien: Im ersten besucht ein niedergelassener Arzt seine Patienten zum Hausbesuch. Im zweiten wird zwischen einer Klinik, dem Patienten und Gesundheitsdienstleistern (z.B. Physiotherapeuten oder Hausärzten) gemeinsam eine Behandlung durchgeführt (siehe [Kit12, Jab13, PTR10, PTKR10] für Details). Abbildung 2.9 zeigt das zweite Szenario mit den Aktivitäten, für die eine mobile Unterstützung gefordert wird.

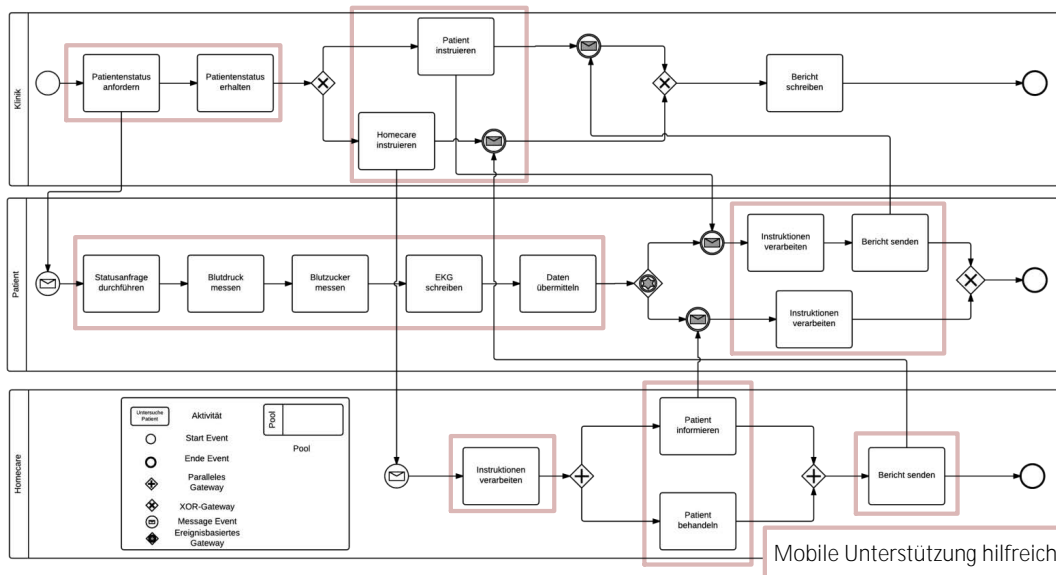


Abbildung 2.9: Prozess zwischen Klinik, Patient und Gesundheitsdienstleister

Telecare-Szenarien sind durch folgende Eigenschaften charakterisiert:

- Es muss von längeren Offline-Zeiten bei der Bearbeitung ausgegangen werden. Die Verbindung kann zudem sehr schlecht sein, sodass die Übertragung von Daten zwischen beteiligten Akteuren sehr lange dauern kann. Ferner muss zwischen den beteiligten Akteuren von unterschiedlichen Netztypen ausgegangen werden.
- Ähnlich wie bei Fallstudie 2 ist die Sensormessung (z.B. Vitalwerte) mit Smart-Mobilgeräten wichtig.
- Eine mobile Unterstützung muss in der Lage sein, Daten zwischenspeichern, wenn keine Online-Verbindung besteht.
- Eine Übergabe laufender Aktivitäten zwischen Smart-Mobilgeräten muss möglich sein. Es kann vorkommen, dass Aufgaben zwischen Akteuren ausgetauscht werden müssen.
- Abhängigkeiten zwischen beteiligten Akteuren müssen definiert werden können.
- Es soll möglich sein, auf einem Smart-Mobilgerät begonnene Aufgaben, auf einem Desktop-System fortzuführen.
- Beim Telecare-Szenario müssen externe Schnittstellen mit IT-Systemen beachtet werden, wenn Smart-Mobilgeräte eingesetzt werden sollen.
- Eine Behandlung von Ausnahmen (z.B. durch Offline-Zeiten, Diskonnektivität) ist auch für das Telecare-Szenario essentiell.

2.3 Anforderungsanalyse

Tabelle 2.7 gibt einen Überblick zu den Anforderungen, die aus der Literaturrecherche und den Fallstudien abgeleitet werden konnten. Wie üblich, wird zwischen funktionalen und nicht-funktionalen Anforderungen unterschieden. Des Weiteren werden die funktionalen Anforderungen in Anforderungen an mobile Aktivitäten und Anforderungen für den mobilen Kontext unterteilt. Tabelle 2.7 zeigt ferner, in welcher Phase der Anforderungsanalyse eine Anforderung identifiziert wurde. Entweder konnte die Anforderung durch die Fallstudie (FS) oder durch die Literaturrecherche (LR) identifiziert werden. Insbesondere ist erkennbar, dass sich jede identifizierte Anforderung aus zumindest einer Fallstudie ableiten ließ. Hingegen ließen sich aus der Literaturrecherche nur etwa die Hälfte der Anforderungen ableiten. Daran lässt sich erkennen, zu welchen Aspekten der in dieser Arbeit vorgestellte Ansatz im Vergleich zu verwandten Ansätzen neue Wege beschreitet.

2.3.1 Funktionale Anforderungen mobiler Aktivitäten

Dieser Abschnitt diskutiert die funktionalen Anforderungen an mobile Aktivitäten.

Anforderung 1 (Kontextabhängige Zuteilung von Aktivitäten) Das *PrMS* muss in der Lage sein, Aktivitäten an mobile Endanwender unter Berücksichtigung von Kontextinformationen zuzuteilen. Dabei müssen zum Beispiel der Ausführungsort der Aktivität und der Aufenthaltsort des Endanwenders berücksichtigt werden können.

Anforderung	Beschreibung	FS	LR
Funktionale Anforderungen mobiler Aktivitäten			
Anforderung 1	Kontextabhängige Zuteilung von Aktivitäten	✓	✓
Anforderung 2	Kontextabhängige Ausnahmebehandlung von Aktivitäten	✓	✓
Anforderung 3	Automatische Priorisierung von Aktivitäten	✓	X
Anforderung 4	Unterbrechen von Aktivitäten	✓	✓
Anforderung 5	Übergabe von Aktivitäten	✓	X
Anforderung 6	Systemwechsel für Aktivitäten	✓	X
Anforderung 7	Spontanes Hinzufügen bzw. Löschen von Aktivitäten	✓	✓
Anforderung 8	Berücksichtigung von Prozess-Constraints	✓	X
Anforderung 9	Unterstützung von Offline-Aktivitäten	✓	X
Funktionale Anforderungen des mobiler Kontexts			
Anforderung 10	Berücksichtigung des Ausführungsorts	✓	✓
Anforderung 11	Berücksichtigung des Netztyps	✓	–
Anforderung 12	Berücksichtigung des Formfaktors	✓	–
Anforderung 13	Berücksichtigung des Energiestatus	✓	–
Anforderung 14	Berücksichtigung von Zeit-Constraints	✓	✓
Anforderung 15	Berücksichtigung des Endanwenderverhaltens	✓	X
Nicht-funktionale Anforderungen mobiler Aktivitäten			
Anforderung 16	Robuste Ausführung	✓	✓
Anforderung 17	Automatische Ausnahmebehandlung	✓	X
Anforderung 18	Frühzeitiges Erkennen von Laufzeitproblemen	✓	X
<i>FS=Fallstudie, LR=Literaturrecherche</i> ✓=trifft zu, X=trifft nicht zu, –=irrelevant			

Tabelle 2.7: Anforderungen an eine robuste Ausführung mobiler Aktivitäten

Anforderung 2 (Kontextabhängige Ausnahmebehandlung von Aktivitäten) Das *PrMS* muss bei der Ausführung mobiler Aktivitäten bei Auftreten von Ausnahmen in der Lage sein, den aktuellen Kontext zu berücksichtigen. Ob eine mobile Aktivität im Ausnahmefall an einen anderen mobilen Endanwender übertragen werden oder eventuell auf einem Desktop-System fortgeführt werden soll, sollte kontextabhängig entschieden werden.

Anforderung 3 (Automatische Priorisierung von Aktivitäten) Das *PrMS* muss in der Lage sein, mobile Aktivitäten für mobile Endanwender zu priorisieren, abhängig vom aktuellen Kontext. So sollte eine Aktivität, von der andere Aktivitäten datenabhängig sind, eine höhere Priorität erhalten, als Aktivitäten, die für andere Aktivitäten keine Daten produzieren.

Anforderung 4 (Unterbrechen von Aktivitäten) Die Ausführung mobiler Aktivitäten muss zu jedem Zeitpunkt unterbrechbar sein. Dies muss sowohl aus Endanwendersicht als auch aus Sicht des *PrMS* möglich sein. Weiters kann dies bedeuten, dass das Smart-Mobilgerät im Fall einer nicht vorhandenen Verbindung Daten zwischenspeichern können muss.

Anforderung 5 (Übergabe von Aktivitäten) Mobile Aktivitäten müssen an andere mobile Endanwender übergeben werden können.

Anforderung 6 (Systemwechsel für Aktivitäten) Die Ausführung einer mobilen Aktivität muss von einem Smart-Mobilgerät auf ein Desktop-System übertragbar sein und umgekehrt.

Anforderung 7 (Spontanes Hinzufügen bzw. Löschen von Aktivitäten) Sofern es der Prozessablauf zulässt, müssen mobile Aktivitäten spontan hinzugefügt oder gelöscht werden

können. Da andere Anforderungen davon aber nicht betroffen sein dürfen, muss der mobile Kontext weiterhin berücksichtigt werden.

Anforderung 8 (Berücksichtigung von Prozess-Constraints) Das *PrMS* muss Prozess-Constraints auch bezogen auf mobile Aktivitäten unterstützen. Ein solcher Constraint könnte lauten, dass zwei Aktivitäten zwingend vom selben mobilen Endanwender ausgeführt werden sollen (sog. Binding of Duties). Bedenkt man, dass ein mobiler Endanwender ausfallen kann, ist die Sicherstellung dieser Anforderung im mobilen Kontext ungleich komplexer sicherzustellen als im stationären Fall. Andere Anforderungen dürfen von dieser Anforderung nicht tangiert werden.

Anforderung 9 (Unterstützung von Offline-Aktivitäten) Das *PrMS* muss in der Lage sein, Aktivitäten auch Offline auszuführen, wenn es der mobile Kontext erfordert. Dies muss wiederum möglich sein, ohne dass eine Ausnahmebehandlung gestartet wird.

2.3.2 Funktionale Anforderungen des mobilen Kontexts

Dieser Abschnitt diskutiert die funktionalen Anforderungen an den mobilen Kontext.

Anforderung 10 (Berücksichtigung des Ausführungsorts) Das *PrMS* muss in der Lage sein, den Ausführungsort einer mobilen Aktivität und den Aufenthaltsort eines mobilen Endanwenders zu berücksichtigen.

Anforderung 11 (Berücksichtigung des Netztyps) Das *PrMS* muss den Netztyp eines Smart-Mobilgeräts berücksichtigen, etwa ob es an einem WLAN oder Telefonnetz angemeldet ist. Der Netztyp wirkt sich wiederum auf den Energieverbrauch und die Übertragungsgeschwindigkeit zwischen *PrMS* und Smart-Mobilgerät aus.

Anforderung 12 (Berücksichtigung des Formfaktors) Das *PrMS* muss den Formfaktor eines Smart-Mobilgeräts berücksichtigen. Zum Betrachten eines Röntgenbildes ist zum Beispiel ein Tablet-PC besser geeignet als ein Smartphone.

Anforderung 13 (Berücksichtigung des Energiestatus) Das *PrMS* muss in der Lage sein, den Batterieladestatus des Smart-Mobilgeräts zu berücksichtigen.

Anforderung 14 (Berücksichtigung einer Dringlichkeit) Das *PrMS* muss in der Lage sein, für mobile Aktivitäten eine Dringlichkeit zu verwalten (z.B. bei Blutabnahmen in einer Klinik).

Anforderung 15 (Berücksichtigung des Endanwenderverhaltens) Das *PrMS* muss spezifisches Verhalten mobiler Endanwender berücksichtigen. Dieser Aspekt wird in verwandten Ansätzen bislang nicht berücksichtigt, seine Relevanz konnte aber in den Fallstudien mehrfach festgestellt werden. So muss das regelmäßige unbedachte Ausschalten des Smart-Mobilgeräts bei einem spontanen Kontextwechsel (z.B. bei einem Notfall in der Klinik) angemessen berücksichtigt werden.

2.3.3 Nicht-funktionale Anforderungen mobiler Aktivitäten

Dieser Abschnitt diskutiert die identifizierten nicht-funktionalen Anforderungen.

Anforderung 16 (Robuste Ausführung) Das *PrMS* muss in der Lage sein, mobile Aktivitäten robust auszuführen. D.h., es muss eine fortschrittliche und leistungsfähige Ausnahmebehandlung zur Verfügung gestellt werden. Diese sollte insbesondere auch das Verhalten des mobilen Endanwenders berücksichtigen.

Anforderung 17 (Automatische Ausnahmebehandlung) Da sich ein mobiler Kontext infolge vieler Aspekte sehr komplex darstellt, muss eine Ausnahmebehandlung automatisch erfolgen. Ein mobiler Endanwender sollte bei der Ausnahmebehandlung im besten Fall nichts manuell entscheiden müssen.

Anforderung 18 (Frühzeitiges Erkennen von Laufzeitproblemen) Das *PrMS* muss in der Lage sein, proaktiv nur solche mobile Endanwender für die Ausführung einer mobilen Aktivität zu berücksichtigen, die weniger wahrscheinlich als andere mobile Endanwender ausfallen. Dabei muss zum Beispiel der Batterieladestatus berücksichtigt werden.

2.4 Zusammenfassung

Dieses Kapitel hat die grundlegenden Anforderungen abgeleitet, die für eine robuste Ausführung mobiler Aktivitäten in einer Prozessumgebung adressiert werden müssen. Insbesondere wurde gezeigt, wie diese ermittelt werden konnten. Dazu wurden einerseits eine umfangreiche Literaturrecherche und andererseits mehrere Fallstudien durchgeführt. Für die abgeleiteten Anforderungen wurde gezeigt, dass sie aktuell meist nicht berücksichtigt werden und der in dieser Arbeit verfolgte Ansatz fortschrittliche Konzepte bietet, die einen originellen Beitrag zum Stand des Wissens leisten. Abschließend kann festgestellt werden, dass die Ableitung der Anforderungen für die robuste Ausführung mobiler Aktivitäten in einer Prozessumgebung sehr differenziert betrachtet werden muss, was in diesem Kapitel im Detail erfolgte.

*You are not a human being in search of a spiritual experience.
You are a spiritual being immersed in a human experience.*

Pierre Teilhard de Chardin (1881-1955)

3

Mobile Prozesse

Der Fokus dieser Arbeit liegt auf der robusten Ausführung mobiler Aktivitäten in einer Prozessumgebung. Die Integration von Smart-Mobilgeräten in eine Prozessausführung führt im Sprachgebrauch zur Bezeichnung des *Mobilen Prozesses*. Ein mobiler Prozess kann jedoch sehr vielfältig definiert werden. Aus diesem Grund zeigt dieses Kapitel auf, welche konkreten Varianten zu unterscheiden sind. Die gewonnenen Erkenntnisse werden genutzt, um die Anforderungsanalyse aus Kapitel 2 zu reflektieren und flankieren. Darüber hinaus sind sie notwendig, um die Stärken und Grenzen des in dieser Arbeit vorgestellten Ansatzes herauszuarbeiten.

Kapitel 3 gliedert sich wie folgt: Abschnitt 3.1 führt in das Thema mobiler Prozesse ein. In Abschnitt 3.2 werden die entscheidenden Aspekte einer Prozessumgebung für die Betrachtung mobiler Prozesse vorgestellt, während Abschnitt 3.3 die Realisierungsvarianten mobiler Prozesse diskutiert. In Abschnitt 3.4 wird diskutiert, ob und - falls ja - welche Rolle die jeweilige Anwendungsdomäne und Wirtschaftlichkeitsaspekte spielen. Abschnitt 3.5 fasst das Kapitel zusammen.

3.1 Einleitung

Mobile Prozesse werden in der Literatur vielfältig definiert (vgl. [RS03, ZHKL10]). Aufgrund der unterschiedlichen Sichtweisen befassen sich entsprechend viele Teildisziplinen der Informatik mit diesem Thema. Ein Exkurs in diese Disziplinen würde den Rahmen dieser Arbeit sprengen. Ein mobiler Prozess wird in diesem Kapitel ausschließlich im Kontext von Business Process Management [PM14] betrachtet. Trotz dieser Fokussierung sind noch viele verschiedene Sichtweisen möglich. Um eine erste Eingrenzung vorzunehmen, wird ein mobiler Prozess als *die Ausführung von Aktivitäten oder Fragmenten eines digitalen Prozesses auf Smart-Mobilgeräten* definiert. Hierbei sind zwei Aspekte entscheidend:

1. Es werden ausschließlich digitale Prozesse (d.h. Workflows) betrachtet. Diese sind in einer Prozessumgebung ausführbar. Ihre Beschreibung wiederum basiert auf wohldefinierten Prozess-Metamodellen bzw. -Ausführungssprachen.

- Wir betrachten die **Ausführung** von Fragmenten eines Prozesses (Aktivität, Aktivitäten oder gesamte Prozesse) auf Smart-Mobilgeräten. Andere Betrachtungen, etwa dazu, ob Fragmente eines Prozesses überhaupt geeignet sind mobil ausgeführt zu werden, werden in dieser Arbeit nicht angestellt.

Ausgehend von obiger Definition mobiler Prozesse lassen sich weitere Formen unterscheiden, die in diesem Kapitel diskutiert werden.

3.2 Prozessumgebung

Die differenzierte Betrachtung der Prozessumgebung, in der mobile Prozesse letztendlich ausgeführt werden, spielt für die Sichtweisen des mobilen Prozesses eine entscheidende Rolle. Die in Tabelle 3.1 dargestellten Aspekte einer Prozessumgebung sind dazu differenziert zu berücksichtigen.

Aspekte der Prozessumgebung	Grund der Betrachtung
Verwendetes <i>PrMS</i> -Architekturkonzept	Je modularer eine Architektur aufgebaut ist (vgl. [Kre14]), desto flexibler können die Konzepte bzw. Module der Prozessumgebung verwendet werden. Sieht die Architektur beispielsweise nicht vor, dass das Protokoll zwischen <i>PrMS</i> und Prozess-Klient flexibel geändert werden kann, stellt das eine erhebliche Einschränkung dar.
Verwendetes <i>Prozess-Klient</i> -Architekturkonzept	Prozess-Klienten sind für die Ausführung von Aktivitäten verantwortlich. Sie laufen auf stationären Systemen bzw. Smart-Mobilgeräten. Dazu teilt das <i>PrMS</i> die Aktivitäten den Prozess-Klienten zu. Für das verwendete Architekturkonzept des Prozess-Klienten ist es entscheidend, ob dieser Klient anfallende Daten zwischenspeichert oder neu erzeugte Daten stets direkt an das <i>PrMS</i> senden muss. Ferner ist zu berücksichtigen, ob ein Prozess-Klient auch Fragmente eines Prozesses autonom ausführen kann (d.h. Prozesslogik eigenständig interpretieren kann).
Verwendetes Klient-Server-Modell zwischen <i>PrMS</i> und <i>Prozess-Klient</i>	In diesem Kontext ist es wichtig, ob nur ein zentrales <i>PrMS</i> existiert oder mehrere, die miteinander kommunizieren. Im letztgenannten Fall ist weiters zu klären, welches <i>PrMS</i> welchem Prozess-Klienten die Aktivitäten zuteilt. Generell existieren viele weitere Klient-Server-Modelle, die sorgfältig zu unterscheiden sind.
Verwendetes Austauschprotokoll zwischen <i>PrMS</i> und <i>Prozess-Klient</i>	Das Protokoll zwischen <i>PrMS</i> und Prozess-Klient ist grundlegend für die Zuteilung von Aktivitäten und die Ausnahmebehandlung. So macht es einen Unterschied, ob das Protokoll auch Zwischenzustände während der Aktivitätsausführung ermöglicht oder ob es nur die beiden folgenden Zustände realisiert: Einen bei der Übergabe der Aktivität und einen nach der Beendigung der Aktivität an den Prozess-Klienten. Ohne weitere Zwischenzustände sind die Möglichkeiten einer Ausnahmebehandlung allerdings erheblich eingeschränkt.

Tabelle 3.1: Wichtige Aspekte der Prozessumgebung

Für den Prozess-Klienten müssen die in Tabelle 3.2 genannten Aspekte im Kontext mobiler Prozesse differenziert berücksichtigt werden.

Aspekte des Prozess-Klienten	Grund der Betrachtung
Nativer mobiler Prozess-Klient	Ein nativer mobiler Prozess-Klient ist ein Prozess-Klient, der auf Smart-Mobilgeräten installiert werden kann. Hinsichtlich Funktionalität entspricht er (meistens) einem stationären Prozess-Klienten.
Browser-basierter mobiler Prozess-Klient	Diese Variante entspricht einem nativen mobilen Prozess-Klienten mit folgendem Unterschied: Es wird keine spezielle Software installiert, vielmehr wird zur Ausführung mobiler Aktivitäten nur ein Browser verwendet, der auf dem Smart-Mobilgerät vorinstalliert ist. Entscheidender Nachteil dieser Lösung ist, dass der Prozess-Klient stets mit dem <i>PrMS</i> verbunden sein muss und nicht alle Hardware-Eigenschaften des Smart-Mobilgeräts genutzt werden können. Die Funktionalität beschränkt sich daher auf diejenigen des Browsers.

Tabelle 3.2: Wichtige Aspekte des mobilen Prozess-Klienten

3.3 Sichtweisen auf mobile Prozesse

Nachfolgend werden verschiedene Sichtweisen auf mobile Prozesse diskutiert. Jede von ihnen muss berücksichtigt werden, wenn mobile Prozesse realisiert werden sollen. Daher spielen die Sichtweisen auch für die robuste Ausführung mobiler Aktivitäten in einer Prozessumgebung eine wichtige Rolle. In der Literatur gibt es bisher keine systematische Gegenüberstellung dieser vier Sichtweisen:

- **Sichtweise 1:** Betrachtung des Aktivitätentyps
- **Sichtweise 2:** Betrachtung des Klient-Server-Modells
- **Sichtweise 3:** Betrachtung des mobilen Prozess-Klienten
- **Sichtweise 4:** Betrachtung von Ad-hoc Änderungen

3.3.1 Sichtweise 1: Betrachtung des Aktivitätentyps

Die erste Sichtweise diskutiert die verschiedenen Varianten automatischer und interaktiver mobiler Aktivitäten (vgl. Abb. 3.1). Eine automatische Aktivität wird ohne manuellen Eingriff von einem Smart-Mobilgerät ausgeführt, interaktive mobile Aktivitäten hingegen werden von einem mobilen Endanwender ausgeführt. In der Literatur existieren nur wenige Ansätze, die sich mit mobilen Aktivitäten beschäftigen. Generell sollten die in Abbildung 3.1 gezeigten vier Varianten unterschieden werden:

- **Variante 1: Einzelne interaktive Aktivitäten** Es sollen einzelne interaktive mobile Aktivitäten auf einem Smart-Mobilgerät ausgeführt werden. Eine solche Aktivitätsausführung wird zum Beispiel von *SAP Netweaver* [SAP14d] unterstützt.
- **Variante 2: Einzelne interaktive Aktivitäten und Fragmente interaktiver Aktivitäten** Es sollen interaktive mobile Aktivitäten auf einem Smart-Mobilgerät ausgeführt werden. Des Weiteren soll das Smart-Mobilgerät auch Fragmente, die aus einer Menge interaktiver mobiler Aktivitäten bestehen, autonom ausführen können (vgl. [PTKR10, que14]).

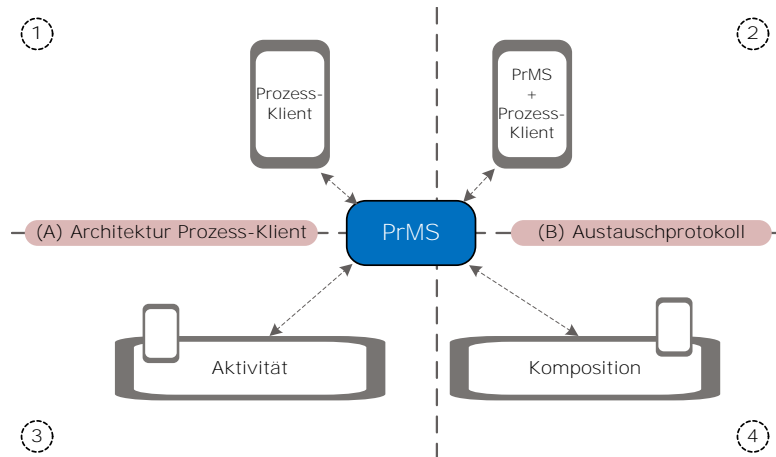


Abbildung 3.1: Automatische und interaktive mobile Aktivitäten

- **Variante 3: Einzelne automatische Aktivitäten** Es wird eine automatische mobile Aktivität auf einem Smart-Mobilgerät ausgeführt. In WS-BPEL [BPE14a] werden solche Aktivitäten zum Beispiel in Form von Web Services, die auf dem Smart-Mobilgerät ablaufen, realisiert [HHGR06].
- **Variante 4: Einzelne automatische Aktivitäten und Fragmente automatischer Aktivitäten** Im Gegensatz zur Variante 3 besteht bei Variante 4 die automatische Aktivität nicht nur aus einer einzelnen mobilen Aktivität, sondern aus einer Komposition mehrerer solcher Aktivitäten. Aus logischer Sicht des *PrMS* wird aber nur eine mobile Aktivität zugeteilt (vgl. [ZL10]). Die konkrete Struktur der Komposition ist für das *PrMS* jedoch nicht ersichtlich. Im Kontext von WS-BPEL wird eine solche Aktivität beispielsweise durch einen WS-BPEL Prozess realisiert (vgl. [ZL10]).

Bezogen auf die vier Varianten spielen zwei Aspekte der Prozessumgebung eine Rolle: (1) die verwendete Architektur des Prozess-Klienten (vgl. Abb. 3.1A) und (2) das Austauschprotokoll zwischen *PrMS* und Prozess-Klient (vgl. Abb. 3.1B). Werden diese beiden Aspekte explizit berücksichtigt, kann beispielsweise entschieden werden, ob das Austauschprotokoll um Zustände für die Ausnahmebehandlung erweitert werden muss. Im Fall automatischer Aktivitäten sind auch weniger Zustände für das Austauschprotokoll denkbar. Oder es muss für das Austauschprotokoll analysiert werden, welches Medium bei der Übertragung typischerweise zum Einsatz kommt (z.B. WLAN vs. Telefonnetz), um dieses explizit zu berücksichtigen. Diese Unterschiede sind für einen Lösungsansatz grundlegend.

In dieser Arbeit wird Variante 1 verfolgt. Die beiden genannten Aspekte der Prozessumgebung werden explizit berücksichtigt. Dennoch wurden die anderen drei Varianten ebenfalls betrachtet, insbesondere um zu entscheiden, ob Teilaspekte dieser Konzepte sinnvoll übernommen werden können.

3.3.2 Sichtweise 2: Betrachtung des Klient-Server-Modells

Dieser Abschnitt diskutiert die verschiedenen Klient-Server-Modelle, die im Kontext mobiler Prozesse zur Anwendung kommen. Nachfolgend werden die verschiedenen Varianten vorgestellt (vgl. Abb. 3.2 und 3.3) und diskutiert.

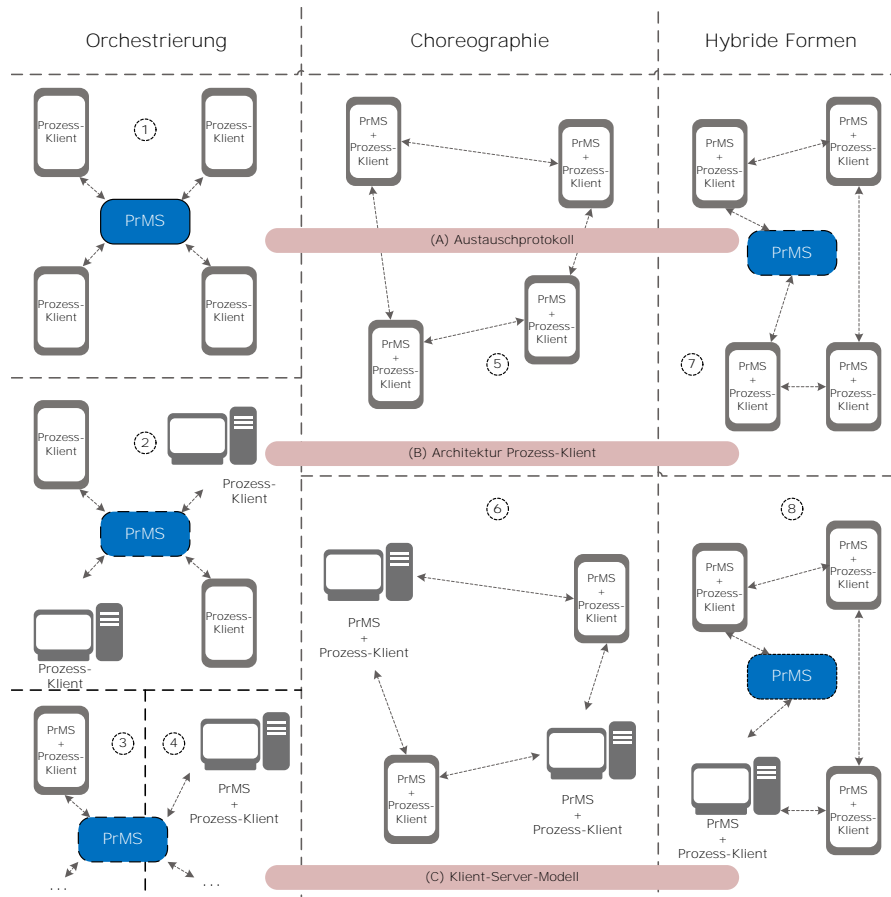


Abbildung 3.2: Orchestrierung und Choreographie mobiler Prozesse

- Variante 1: Orchestrierung** Das verbreitetste Klient-Server-Modell ist die Orchestrierung (vgl. Abb. 3.2①-④). Bei diesem Modell teilt ein zentrales *PrMS* die Aktivitäten den Prozess-Klienten zu. Weiters müssen vier Sub-Varianten unterschieden werden: (1) Es werden nur mobile Prozess-Klienten verwendet (vgl. Abb. 3.2①). (2) Es werden nicht nur mobile, sondern auch stationäre Prozess-Klienten verwendet (vgl. Abb. 3.2②). (3) Es werden nur mobile Prozess-Klienten verwendet; diese können ferner Fragmente autonom ausführen (vgl. Abb. 3.2③). (4) Es werden nicht nur mobile, sondern auch stationäre Prozess-Klienten verwendet; diese sind in der Lage, Fragmente autonom auszuführen (vgl. Abb. 3.2④). WS-BPEL [BPE14a] ist ein Beispiel für eine Orchestrierungssprache.
- Variante 2: Choreographie** Die Choreographie ist ein weiteres Klient-Server-Modell (vgl. Abb. 3.2⑤,⑥). Bei dieser Variante existiert kein zentrales *PrMS* und die Prozess-Klienten realisieren einen dezentralen Austausch des Gesamtprozesses. Für den Austausch existieren zwei grundlegend verschiedene Ansätze. Entweder wird die

komplette Prozessaufführung von Prozess-Klient zu Prozess-Klient weitergereicht (sog. Instanz-Migration; vgl. Abb. 3.3②). Oder ein Prozess wird vor der Ausführung fragmentiert und die resultierenden Fragmente werden auf die Prozess-Klienten verteilt (sog. Fragmentierung; vgl. Abb. 3.3①). Ferner muss ein Synchronisationsprotokoll zwischen den ausführenden Prozess-Klienten realisiert werden.

Weiters müssen zwei Sub-Varianten unterschieden werden, die wiederum mit den Choreographiearten kombiniert auftreten können: (1) Es werden nur mobile Prozess-Klienten verwendet (vgl. Abb. 3.2⑤). (2) Es werden nicht nur mobile, sondern auch stationäre Prozess-Klienten verwendet (vgl. Abb. 3.2⑥). WS-CDL [WSC14] ist ein Beispiel für eine Choreographiesprache. Choreographien spielen im Kontext unternehmensübergreifender Prozesse [KRP⁺13, KRL⁺13, KPR12b, KPR12a, KRFRM13] eine wichtige Rolle.

- **Variante 3: Hybrid** Außerdem existieren hybride Formen der Varianten 1 und 2. So kann ein zentrales *PrMS* existieren, Prozess-Klienten aber untereinander kommunizieren (vgl. Abb. 3.2⑦). Bei letztgenannter Variante können zudem nicht nur mobile Prozess-Klienten verwendet, sondern auch stationäre Prozess-Klienten hinzugenommen werden; dann resultiert eine weitere hybride Form (vgl. Abb. 3.2⑧).

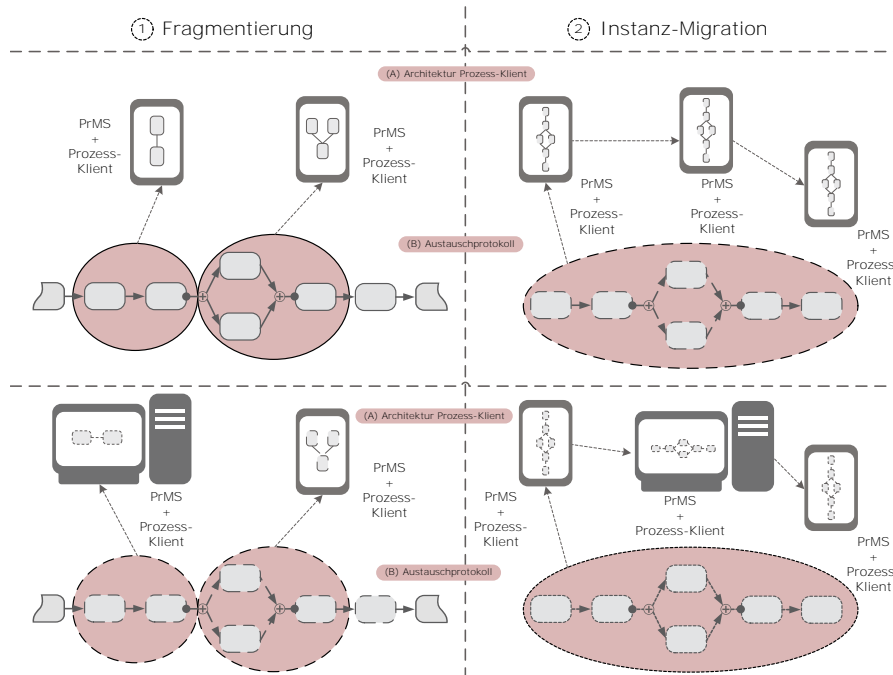


Abbildung 3.3: Fragmentierung und Instanz-Migration

Eine detaillierte Diskussion dieser Varianten würde den Rahmen dieser Arbeit sprengen. Im Folgenden werden daher nur die wichtigsten Aspekte zusammengefasst.

- Generell erfordert eine Choreographie im Hinblick auf die Realisierung mobiler Prozesse viele Überlegungen bezüglich der Prozessumgebung (vgl. Abb. 3.2(A,B,C)). So muss beachtet werden, dass die Verbindung zwischen den Prozess-Klienten als grundsätzlich instabil angesehen werden muss und abhängig vom Szenario variieren kann. Ferner können etablierte Protokolle zwischen *PrMS* und Prozess-Klient nicht übernommen werden,

da die Prozess-Klienten direkt miteinander kommunizieren müssen. Gleiches gilt für die Architektur der Prozess-Klienten. Auch muss sorgsam entschieden werden, ob bei den Prozess-Klienten nur Smart-Mobilgeräte eingesetzt werden oder stationäre Systeme zusätzlich zum Einsatz kommen sollen.

- Die Instanz-Migration stellt die am meisten verbreitete Variante zur Realisierung mobiler Prozesse in der Literatur dar (vgl. [ZHKL10]). Im Kern muss hier ein Kriterium bestimmt werden, welches festlegt, wann die Prozessaufführung von einem Prozess-Klienten an den nächsten weitergegeben wird. Da eine manuelle Festlegung zur Ausführungszeit zu komplex wäre, sollte das Kriterium eine automatische Weitergabe ermöglichen. Kann ein solches Kriterium gefunden werden, kann die Prozessaufführung flexibel zwischen Prozess-Klienten weitergereicht werden oder sogar mehrere Prozess-Klienten parallel laufen. Bisher konzentrieren sich verwandte Ansätze auf eine Prozessumgebung, die technisch in der Lage ist, solch ein Weitergeben auch auf Smart-Mobilgeräten zu ermöglichen (vgl. [ZHKL10]).

Das konkrete Kriterium für die Zuteilung von Fragmenten auf Smart-Mobilgeräte wird in den meisten verwandten Ansätzen kaum betrachtet. Jedoch stellt ein solches Kriterium die wichtigste Komponente mobiler Prozesse auf Basis einer Instanz-Migration dar. Im besten Fall sollte dies den mobilen Kontext berücksichtigen, um eine robuste Ausführung zu gewährleisten. Außerdem wird in verwandten Ansätzen wenig berücksichtigt, dass bei der Weitergabe zwischen Prozess-Klienten ein Smart-Mobilgerät einerseits instabiler ist als ein stationäres System und andererseits nicht jedes Smart-Mobilgerät die gleichen Eigenschaften vorweist. Bei einer Instanz-Migration, die nur zwischen stationären Prozess-Klienten weitergegeben wird, spielen diese Aspekte keine Rolle.

- Bei der Fragmentierung spielt das Kriterium zur Bestimmung der Fragmente eine wichtige Rolle. Hier jedoch werden die Fragmente bereits vor der Ausführung zu Prozess-Klienten zugeordnet, weshalb die Festlegung des Kriteriums vereinfacht wird. Dafür kann während der Ausführung der Prozess-Klient, der das Fragment ausführt, nicht mehr geändert werden. Durch diese fehlende Ausführungsflexibilität verfolgen nur wenige Ansätze diese Variante zur Realisierung mobiler Prozesse.
- Sowohl bei der Instanz-Migration als auch der Fragmentierung spielen die Architektur des Prozess-Klienten sowie die Austauschprotokolle eine große Rolle. Man betrachte hierzu folgendes Beispiel: Bei der Instanz-Migration kann es vorkommen, dass ein Smart-Mobilgerät mehr als ein Fragment ausführt. In einem solchen Fall muss überlegt werden, ob dies im Austauschprotokoll berücksichtigt werden muss. So könnte eine Maßnahme sein, Teile der Ausführungsdaten lokal auf dem Smart-Mobilgerät zwischenspeichern, sodass das Austauschprotokoll nicht immer alle Daten übertragen muss, da aus einer vorherigen Ausführung bereits Ausführungsdaten bekannt sind.
- Obwohl die Orchestrierung viele Aspekte der Choreographie zur Realisierung mobiler Prozesse vereinfacht, etwa dass zum *PrMS* immer eine stabile Verbindung vorausgesetzt werden kann, dürfen auch hier die Unterschiede der möglichen Realisierungsformen nicht vernachlässigt werden. So ist es ein großer Unterschied, ob nur Smart-Mobilgeräte als Prozess-Klienten verwendet werden oder stationäre Systeme zusätzlich zum Einsatz kommen sollen. Im letztgenannten Fall sollten beide Arten von Prozess-Klienten mit spezifischen Austauschprotokollen zwischen *PrMS* und Smart-Mobilgerät bedacht werden. Außerdem lassen hier auch die Ausnahmebehandlungsmöglichkeiten differenzierter be-

trachten, da beispielsweise während der Ausführung einer Aktivität von einem Smart-Mobilgerät auf ein stationäres System gewechselt werden kann.

In der vorliegenden Arbeit wird **Variante 1** und damit **die Orchestrierung als Klient-Server-Modell** gewählt. Außerdem wird für die **Orchestrierung die Sub-Variante 2** gewählt, die als Prozess-Klienten sowohl Smart-Mobilgeräte als auch stationäre Systeme zulässt. Obwohl in der vorliegenden Arbeit nur die Orchestrierung als Klient-Server-Modell relevant ist, soll an einem Beispiel diskutiert werden, welche Rolle die anderen Varianten spielen könnten. Da das Austauschprotokoll als ein entscheidender Faktor der Prozessumgebung identifiziert wurde, haben wir für die entwickelten Konzepte die Austauschprotokolle von Choreographie- und Orchestrierungsvarianten miteinander verglichen. Dabei stellte sich heraus, dass Optimierungen nur im Choreographiefall diskutiert werden. Diese Optimierungen spielen wiederum im Kontext einer Ausnahmebehandlung im Orchestrierungsfall eine Rolle.

3.3.3 Sichtweise 3: Betrachtung des mobilen Prozess-Klienten

Die dritte Sichtweise diskutiert zwei Varianten für die Realisierung eines mobilen Prozess-Klienten:

- **Variante 1: Nativer Prozess-Klient** Es wird ein sog. nativer mobiler Prozess-Klient realisiert. Nativ heißt, dass der Prozess-Klient spezifisch für ein mobiles Betriebssystem (z.B. iOS, Android oder Windows Phone) entwickelt wird. Die Programmiersprache, die vom Hersteller zur Verfügung gestellt wird, muss zwingend für einen nativen mobilen Prozess-Klienten verwendet werden.
- **Variante 2: Browser-basierter Prozess-Klient** Es wird ein sog. Browser-basierter mobiler Prozess-Klient realisiert. In diesem Fall wird ein Browser auf dem Smart-Mobilgerät dazu verwendet, die Funktionalität des Prozess-Klienten zu realisieren. Die Möglichkeiten zur Programmierung des Prozess-Klienten sind im Gegensatz zur Variante 1 in diesem Fall vielfältig.

Zu den Varianten kommt ein weiterer Aspekt hinzu, den es zu berücksichtigen gilt: Die Entwicklung von Softwareanwendungen für Smart-Mobilgeräte unterscheidet sich in mehrerlei Hinsicht von der Entwicklung von Softwareanwendungen für stationäre Systeme. Eine tiefergehende Behandlung würde den Rahmen dieser Arbeit sprengen. Die Unterschiede haben in der Vergangenheit dazu geführt, dass neue Konzepte entwickelt wurden, um Softwareanwendungen für Smart-Mobilgeräte zu entwickeln [SSP⁺13, IBM14b]. Diese Konzepte müssen auch für die Realisierungsvarianten mobiler Prozess-Klienten berücksichtigt werden.

Da die Konzepte mobiler Anwendungsentwicklung einen großen Einfluss auf die Realisierungsvarianten mobiler Prozess-Klienten haben können, werden diese zuerst diskutiert. Generell unterscheidet man die folgenden Konzepte zur Entwicklung mobiler Anwendungen (vgl. [IBM14b, SRP⁺15]):

- **Web-Anwendung:** Es wird der Browser auf dem Smart-Mobilgerät verwendet, um auf eine Web-Anwendung zuzugreifen. Entscheidend für diese Variante ist, dass die Web-Anwendung physisch nicht auf dem Smart-Mobilgerät liegt und daher eine Online-Verbindung, etwa zu einem Server im Internet, notwendig ist. Ferner kann eine Web-Anwendung nicht als sog. App im App-Store [APP14a] verfügbar gemacht werden.

- **Mobile Web-Anwendung:** Die mobile Web-Anwendung kann im Gegensatz zur Web-Anwendung lokal auf dem Smart-Mobilgerät ausgeführt werden. Erst wenn zum Beispiel Daten mit einem Server ausgetauscht werden sollen, muss eine Online-Verbindung hergestellt werden.
- **Hybride mobile Anwendung:** Mithilfe sog. Cross-Frameworks (vgl. [CF14, Cor14, IBM14d, App14b, STo14]) werden hybride mobile Anwendungen entwickelt. Charakteristisch für dieses Konzept mobiler Anwendungsentwicklung sind drei Aspekte:
 1. Es werden bewusst nicht native mobile Programmiersprachen verwendet, um eine mobile Anwendung zu entwickeln. Dies hat den Vorteil, dass Programmierer die native Programmiersprache nicht erlernen müssen. Meistens kommen verbreitete Programmiersprachen (z.B. Java) zum Einsatz. Die verwendete Programmiersprache des Cross-Frameworks wird als Intermediate-Sprache bezeichnet.
 2. Cross-Frameworks sind so konzipiert, dass die in der Intermediate-Sprache realisierte mobile Anwendung in alle gängigen nativen mobilen Programmiersprachen übersetzt werden kann (sog. Plattformunabhängigkeit). Somit ist gewährleistet, dass aus einer Code-Basis mobile Anwendungen für alle gängigen mobilen Betriebssysteme realisiert werden können.
 3. Eine hybride mobile Anwendung kann stets im App-Store [APP14a] verfügbar gemacht werden.

Diese Art der mobilen Anwendungsentwicklung erscheint effektiv. Jedoch muss ein Aspekt angemerkt werden: In der Hauptsache erreichen Cross-Frameworks die Plattformunabhängigkeit dadurch, dass die Intermediate-Sprache in mobile Web-Anwendungen übersetzt wird und eine sog. Container-Anwendung die Web-Anwendung zu einer App *verpackt*. Somit gelten für die hybriden mobilen Anwendungen viele Merkmale mobiler Web-Anwendungen.

- **Native mobile Anwendung:** Für eine native mobile Anwendung wird ausschließlich die vom Betriebssystemhersteller zur Verfügung gestellte Programmiersprache genutzt. Nur mit diesem Konzept können alle Eigenschaften der Smart-Mobilgeräte (z.B. gerätespezifische Sensorik) verwendet werden. Ferner ist dadurch auch die bestmögliche Usability [Sch11a] gewährleistet, was in vielen Szenarien eine entscheidende Rolle spielt.

Varianten mobiler Prozess-Klienten und Konzepte mobiler Anwendungsentwicklung

Um einen nativen mobilen Prozess-Klienten zu entwickeln, muss eine native mobile Anwendung realisiert werden. Dies entspricht der Variante 1 eines mobilen Prozess-Klienten. Für die Variante 2 des Browser-basierten Prozess-Klienten kommen die übrigen drei Konzepte mobiler Anwendungsentwicklung infrage.

Tabelle 3.3 fasst die Vor- und Nachteile der genannten Konzepte zusammen¹.

¹Zur Vertiefung dieser Aspekte dienen folgende Quellen: [RPR11, GPSR13, PLRH12, GSP⁺14, PMLR14, SSP⁺13].

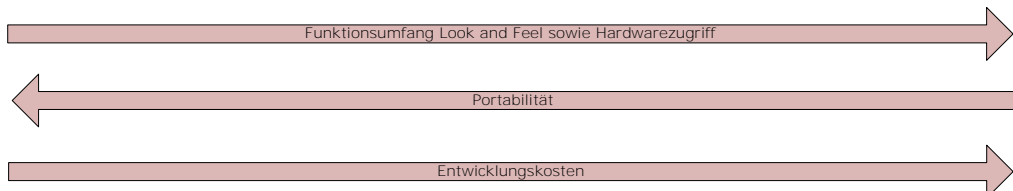
Vergleichs- aspekte	Web- Anwendung	Mobile Web- Anwendung	Hybride mobile Anwendung	Native mobile Anwendung
Programmiersprache	HTML5 Javascript	HTML5 Javascript	Intermediate- Sprachen	Plattform- spezifisch Java, Objective- C, ..
Vorteile	(1) Plattform-unabhängig (2) niedrige Entwicklungskosten (3) Sicherheit gut realisierbar (4) Berücksichtigung aller Formfaktoren	(1) Offline-fähig (2) siehe Web-Anwendung	(1) siehe Web-Anwendung (2) performanter als Web- Anwendungen (3) Im App-Store verfügbar (4) Guter Hardwarezugriff (5) Offline-fähig	(1) gute Performanz (2) Im App-Store [APP14a] verfügbar (3) gutes Look and Feel
Nachteile	(1) Eingeschränkter Hardware- zugriff (2) Nicht in App-Store verfügbar (3) Schlechte Performanz (4) Schlechte Skalierbarkeit (5) Schlechtes Look and Feel	(1) siehe Web-Anwendung	(1) Updates Betriebssystem explizit berücksichtigen (2) schlechtere Performance als Web- Anwendungen (3) schlechtes Look and Feel	(1) Plattform- abhängig (2) Hohe Entwicklungskosten (3) Hohe Wartungskosten (4) Formfaktor explizit berücksichtigen
Deployment	Läuft im Browser		App-Store Deployment	
				

Tabelle 3.3: Vor- und Nachteile mobiler Anwendungsentwicklung im Überblick

Um die genannten Varianten aus Entwicklersicht zu beurteilen, wurden umfangreiche Forschungs- und Entwicklungsprojekte durchgeführt, in denen alle Konzepte mobiler Anwendungsentwicklung zum Einsatz kamen (vgl. Tabelle 3.4).

Mithilfe der durchgeführten Projekte konnte der in Tabelle 3.5 gezeigte Vergleichskatalog für einen mobilen Anwendungsentwickler aufgestellt werden.

Generell lässt sich aus den durchgeführten Projekten das Folgende festhalten: Die Grundsatzfrage lautet stets, ob eine Anwendung nativ oder mit einer der anderen drei Varianten realisiert werden soll. Unter Berücksichtigung der in den genannten Projekten gemachten Erfahrung sprechen drei Gründe für die Entwicklung nativer Anwendungen: (1) Eine umfangreiche Ver-

Projekt	Web	Hybrid	Nativ	Bemerkungen
<i>AREA</i>	<i>X</i>	<i>X</i>	✓	Verknüpfung von Augmented Reality mit Prozess-Technologie [GSP ⁺ 14, GPSR13, SPSR15]. Sensor-Zugriffe erfordern native Entwicklung.
<i>DBIScholar</i>	<i>X</i>	<i>X</i>	✓	Berechnung von Zitierindexen [RPR11, Rob11] auf Basis von Google Scholar [GS14]. Aufwändiges Parsen großer Datenmengen erfordert native Entwicklung.
<i>MeDO</i>	<i>X</i>	<i>X</i>	✓	Mobile Anwendung zur Unterstützung von Ärzten während der Visite [Lan12, PMLR14, PLRH12]. Visite erfordert spezialisierte Benutzerschnittstelle und damit native Entwicklung.
<i>SnapOS</i>	<i>X</i>	✓	<i>X</i>	App für Gastronomie-Betriebe zur Bestellung von Speisen. Einsatz des Cross-Frameworks PhoneGap [PG14].
<i>Orianda</i>	<i>X</i>	✓	<i>X</i>	Wartung von Schienen der Deutschen Bahn unter Einsatz von Smart-Mobilgeräten. Einsatz des Cross-Frameworks Sybase Unwired Platform [SUP14] von SAP.
<i>MobileCMS</i>	✓	✓	<i>X</i>	Einsatz von PhoneGap [PG14] und Responsive Web-Techniken [RWE14], um mobile Apps gegen ein aktuelles Web-Content-Management-System [Web14] zu realisieren [Sta14].
<i>TrackYourTinnitus</i>	<i>X</i>	<i>X</i>	✓	Projekt zur Überwachung von Tinnitusgeschädigten Patienten [SHP ⁺ 14a, SHP ⁺ 14b, Her14, PRH ⁺ 15, PRLS15]. Viele Hardware-Funktionen des Smart-Mobilgeräts müssen angesprochen werden, erfordert native Entwicklung.
<i>QuestionSys</i>	<i>X</i>	<i>X</i>	✓	Entwicklung mobiler Fragebögen zur Unterstützung klinischer Studien der Psychologie [SSP ⁺ 14, SRLP ⁺ 13, SPR15, SSPR15]. Aufwändige Benutzerschnittstelle erfordert native Entwicklung.

Tabelle 3.4: Forschungsprojekte im Kontext mobiler Anwendungsentwicklung

wendung der Hardwaremerkmale des Smart-Mobilgeräts; (2) Notwendigkeit einer aufwändigen Benutzerschnittstelle und (3) Notwendigkeit einer performanten mobilen Anwendung. Trifft einer dieser Gründe zu, sollte die mobile Anwendung nativ realisiert werden, in allen anderen Fällen muss abhängig vom Szenario die beste Variante gewählt werden.

Die gewonnenen Erkenntnisse werden nachfolgend im Kontext mobiler Aktivitäten diskutiert. Abbildung 3.4 stellt die Ausführung mobiler Aktivitäten mit einem nativen mobilen Prozess-Klienten sowie mithilfe eines Browser-basierten mobilen Prozess-Klienten gegenüber.

Folgende Varianten sind in Abbildung 3.4 dargestellt:

Entwicklungsaspekt	Web	Hybrid	Nativ
Entwicklungsaufwand	Mittel	Mittel	Sehr Hoch
Entwicklungszeit	Kurz	Kurz	Lang
Anwendungsportabilität	Hoch	Hoch	Gering (oder gar nicht)
Performanz	Schnell	Schnell (besser als bei Web)	Sehr schnell
Zusatzfunktionen nutzbar	Nein	Teilweise	Alle
Deployment über App-Store	Nein	Teilweise	Ja
Natives Look and Feel	Eingeschränkt	Eingeschränkt (besser als bei Web)	Ja

Tabelle 3.5: Vergleich der Konzepte mobiler Anwendungsentwicklung

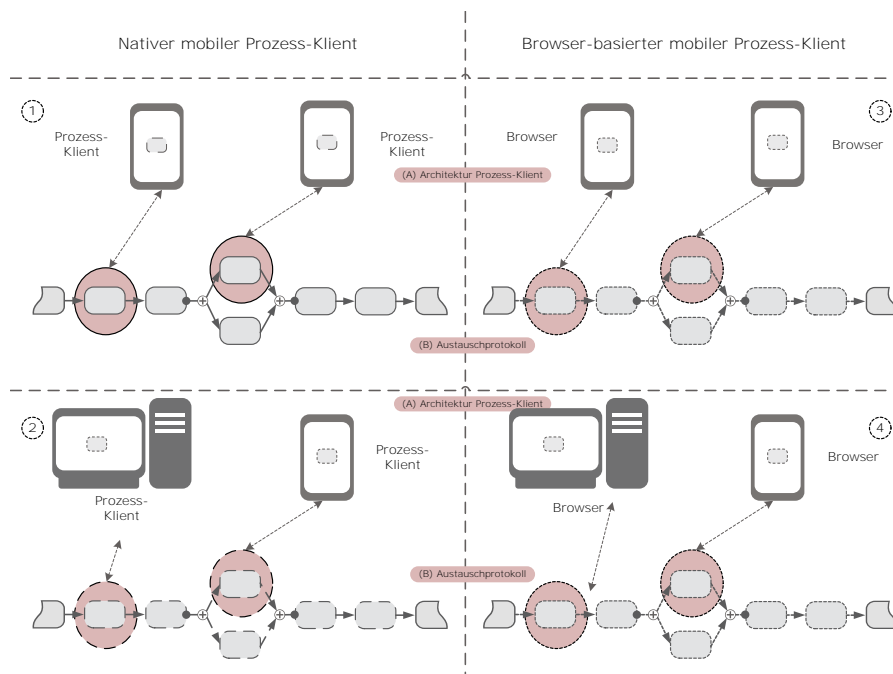


Abbildung 3.4: Nativer und Browser-basierter mobiler Prozess-Klient

- **Variante 1: Nativer Prozess-Klient (nur mobil)** Diese Variante sieht nur native mobile Prozess-Klienten vor.
- **Variante 2: Nativer Prozess-Klient (mobil und stationär)** Diese Variante sieht sowohl native mobile als auch stationäre Prozess-Klienten vor.
- **Variante 3: Browser-basierter Prozess-Klient (nur mobil)** Diese Variante sieht nur Browser-basierte mobile Prozess-Klienten vor.
- **Variante 4: Browser-basierter (mobil und stationär)** Diese Variante sieht sowohl Browser-basierte mobile Prozess-Klienten als auch stationäre Prozess-Klienten vor.

Eine weitere Möglichkeit, die nicht in Abbildung 3.4 dargestellt wird, besteht darin, native und Browser-basierte mobile Prozess-Klienten zu kombinieren. Jedoch spielt dieser Unterschied für die folgenden Überlegungen keine Rolle. Weiters ist in Abbildung 3.4 (siehe A und B) zu erkennen, dass sowohl die Architektur des Prozess-Klienten als auch das Austauschprotokoll für die in dieser Arbeit entwickelten Konzepte eine Rolle spielen. Mit einem nativen mobilen Prozess-Klienten lassen sich diese Aspekte bestmöglich adressieren. In den von uns durchgeführten Arbeiten [Sta14, Mus14] konnte zudem gezeigt werden, dass auch mit einer Browser-basierten Variante eine Umsetzung möglich ist.

Abschließend lässt sich festhalten, dass die diskutierten Varianten der Sichtweise 3 und die Diskussion zur mobilen Anwendungsentwicklung nur eine untergeordnete Rolle für die Konzepte dieser Arbeit darstellen. Lediglich eine Einschränkung muss gemacht werden: Web-Anwendungen, die eine ständige Online-Verbindung voraussetzen, sind für die Ziele dieser Arbeit nicht verwendbar. Dies stellt jedoch nur eine geringe Einschränkung dar, da mobile Web-Anwendungen gleichwertig zu Web-Anwendungen verwendet werden können.

Obwohl, wie gezeigt, für die Ziele dieser Arbeit fast alle Varianten mobiler Anwendungsentwicklung möglich sind, wurde für die prototypische Realisierung auf einen nativen mobilen Prozess-Klienten zurückgegriffen. Da die entwickelten Konzepte intensiven Gebrauch einer lokalen Zwischenspeicherung auf dem Smart-Mobilgerät machen, bietet sich hierzu ein nativer mobiler Prozess-Klient an. Die Hersteller mobiler Betriebssysteme unterstützen den Entwickler zudem bei hardwarenahen Funktionen gut. So lassen sich zum Beispiel Threads [Thr14] besser für einen nativen Prozess-Klienten programmieren, was wiederum für die Realisierung der Ausnahmebehandlung mobiler Aktivitäten eine wichtige Rolle spielt. Daher fiel die Wahl auf einen nativen mobilen Prozess-Klienten.

3.3.4 Sichtweise 4: Betrachtung von Ad-hoc Änderungen

Eine weitere Sichtweise ergibt sich für den Fall von Ad-hoc Prozess-Änderungen mobiler Aktivitäten. Letztere werden in [Rei00] diskutiert und Lösungen zur Realisierung vorgestellt. Ad-hoc Änderungen ermöglichen es zur Ausführungszeit eines Prozesses, einzelne Aktivitäten oder Fragmente spontan hinzuzufügen oder zu löschen, ohne jedoch die korrekte Ausführbarkeit des Prozesses nach seiner Änderung zu verletzen. Für mobile Aktivitäten sind Ad-hoc Änderungen wichtig, da in einem mobilen Kontext die Anforderung spontaner Änderungen relativ wahrscheinlich ist. Zum Beispiel muss bei einem Patienten in einer Klinik, der spontan zum Notfall wird, der Blutzucker gemessen werden können.

Entsprechend der Literatur sind zwei Varianten von Ad-hoc Änderungen zu unterscheiden (vgl. Abb. 3.5):

- **Variante 1: Änderung des Ausführungsschemas** Während der Prozessausführung werden weitere mobile Aktivitäten (interaktive oder automatische) hinzugefügt oder gelöscht. Diese Variante stellt die *klassische* Ad-Hoc Änderung dar [Rei00]. Diese ist ferner dadurch gekennzeichnet, dass ein *PrMS* die jeweiligen Änderungen zentral realisiert.
- **Variante 2: Änderung des Smart-Mobilgeräts** Hier wird nicht das Ausführungsschema des Prozesses durch Hinzufügen oder Löschen von Aktivitäten spontan geändert, sondern das ausführende Smart-Mobilgerät einer mobilen Aktivität spontan gewechselt. Die Ad-hoc Änderung ist ferner dadurch gekennzeichnet, dass sie nicht vom *PrMS* zentral, sondern mittels Peer-to-Peer Protokoll [P2P14], direkt zwischen den ausführenden

Smart-Mobilgeräten durchgeführt wird [SDGH09, Sch09, SKHR08]. Meistens werden solche Ad-hoc Änderungen dazu verwendet, auf Smart-Mobilgeräte mit besseren Ausführungsbedingungen (z.B. wegen einer besseren Akkuleistung) zu wechseln.

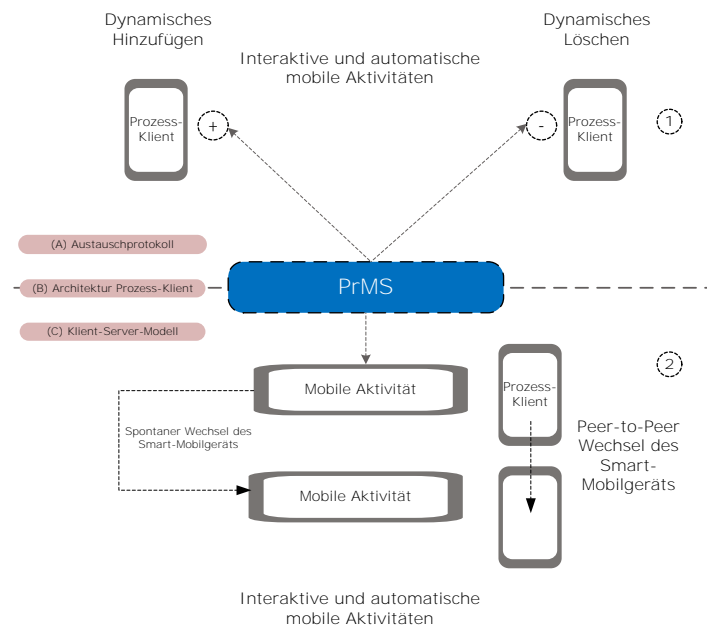


Abbildung 3.5: Ad-hoc Änderungen mobiler Aktivitäten

Für beide Varianten sind das Austauschprotokoll und die Architektur des Prozess-Klienten entscheidend. Insbesondere die Architektur des Prozess-Klienten spielt für Variante 2 eine wichtige Rolle. Wenn der Prozess-Klient dynamisch migriert werden können soll, muss die Architektur eine solche Maßnahme geeignet unterstützen.

Variante 1 wird in fast allen verwandten Ansätzen im Kontext mobiler Prozesse diskutiert, Variante 2 hingegen nur in wenigen verwandten Ansätzen betrachtet. Eine Kombination beider Varianten konnte bisher nicht gefunden werden. Im vorliegenden Ansatz wird Variante 1 realisiert, in der mobile Aktivitäten spontan hinzugefügt oder gelöscht werden können. Obwohl die Konzepte dieser Arbeit auch einen Wechsel des Smart-Mobilgeräts zulassen, wird dieser mithilfe des *PrMS* zentral realisiert und entspricht daher Variante 1. Die Variante 2 eines Peer-to-Peer Wechsels zwischen Smart-Mobilgeräten wird bislang nicht realisiert und würde umfangreiche weitere Überlegungen notwendig machen.

3.4 Diskussion

Abschließend soll noch ein weiterer Aspekt genannt werden, der im Kontext der diskutierten Sichtweisen mobiler Prozesse relevant ist. Um eine breit nutzbare Lösung zu erzielen, sollten stets unterschiedliche Anwendungsdomänen betrachtet werden, da jede von ihnen ggf. unterschiedliche Anforderungen an mobile Prozesse hat. Um dies zu verdeutlichen, werden die beiden Prozesse aus Abbildung 3.6 betrachtet. Der erste Prozess (vgl. Abb. 3.6①) zeigt die mobilen Aktivitäten einer Kreditbearbeitung und der zweite (vgl. Abb. 3.6②) die mobilen Aktivitäten

einer OP-Durchführung. Tabelle 3.6 führt zu beiden Prozessen an, welche Aspekte der vorgestellten Sichtweisen eine Rolle für diese beiden Prozesse spielen. Wie zu sehen ist, ergeben diese beiden Prozesse ein unterschiedliches Resultat bezüglich der Sichtweisen.

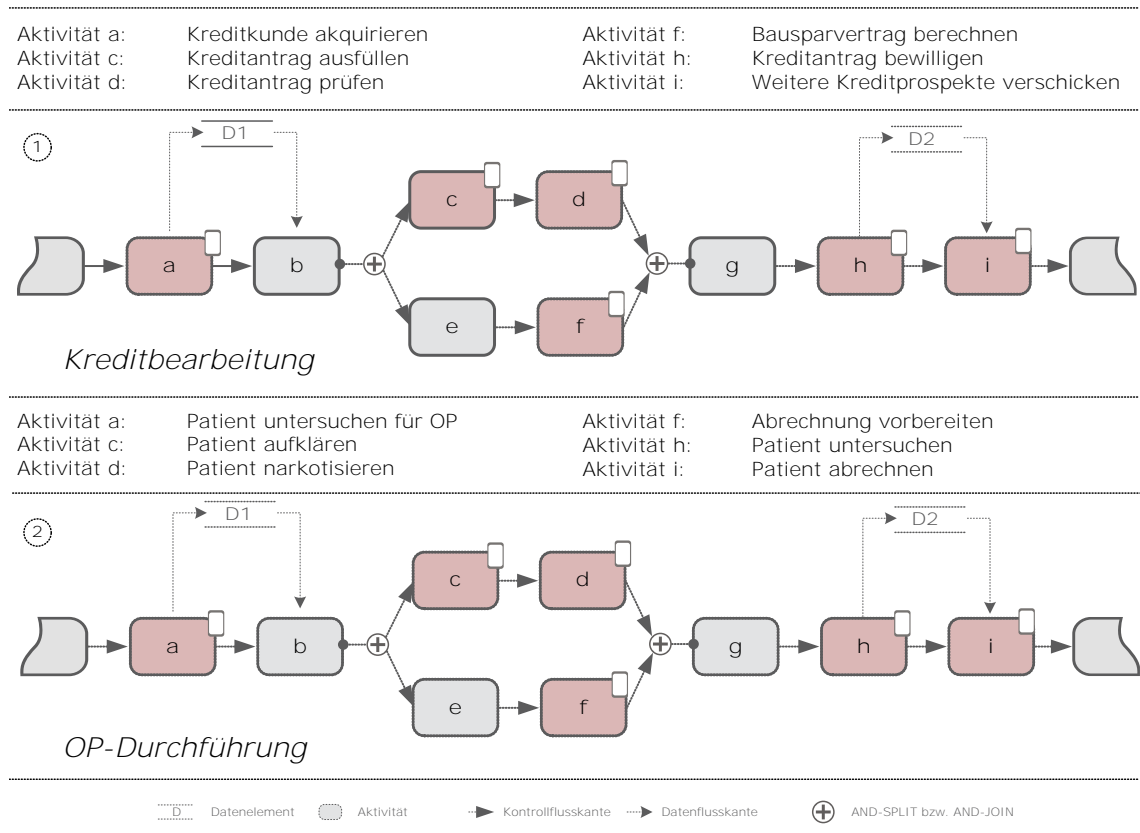


Abbildung 3.6: Beispielprozesse verschiedener Anwendungsdomänen

	Kreditbearbeitung	OP-Durchführung
Sichtweise 1 Aktivitätentyp	Variante 1 Einzelne interaktive Aktivitäten	Variante 2 Einzelne interaktive Aktivitäten und Fragmente interaktiver Aktivitäten
Sichtweise 2 Klient-Server-Modell	Variante 1 Orchestrierung	Variante 3 Hybrid
Sichtweise 3 Mobiler Prozess-Klient	Variante 2 Browser-basierter mobiler Prozess-Klient	Variante 1 Nativer mobiler Prozess-Klient
Sichtweise 4 Ad-Hoc Änderungen	Ad-hoc Änderungen sind nicht notwendig	Variante 1 Änderung des Ausführungsschemas und Variante 2 Änderung des Smart-Mobilgeräts

Tabelle 3.6: Sichtweisen und Prozessbeispiele

3.5 Zusammenfassung

Die Möglichkeiten zur Definition bzw. Realisierung mobiler Prozesse sind vielfältig. Dieses Kapitel hat herausgearbeitet, dass die Prozessumgebung eine entscheidende Rolle spielt und zudem differenziert betrachtet werden muss. Ausgehend davon wurde diskutiert, welche Sichtweisen für mobile Prozesse eingenommen werden sollten. Für jede Sichtweise wurde gezeigt, welche Rolle die Prozessumgebung spielt. Weiters wurden für jede Sichtweise Varianten vorgestellt, und es wurde ausgeführt, welche dieser Varianten für die vorliegende Arbeit verfolgt wird. Abschließend wurde verdeutlicht, dass die Anwendungsdomäne bei der Realisierung mobiler Prozesse ebenfalls wichtig ist.

Die Fesseln der Gewohnheit sind meist so fein, dass man sie gar nicht spürt. Doch wenn man sie dann spürt, sind sie schon so stark, dass sie sich nicht mehr zerreißen lassen.

Samuel Johnson (1709-1784)

4

Stand der Technik

Bereits in Kapitel 3 wurde gezeigt, dass eine Verknüpfung von Mobilität und Prozessen aus verschiedenen Perspektiven betrachtet werden sollte. Davon ausgehend sowie unter Einbezug der in Kapitel 2 diskutierten Anforderungen, lassen sich nun verwandte Ansätze, Forschungsprojekte und Standards identifizieren, die im Kontext dieser Arbeit relevant sind. Diese werden in diesem Kapitel diskutiert. Auf Grundlage von Kapitel 2 und 3 ließen sich sechs Kategorien relevanter Ansätze identifizieren (vgl. Tabelle 4.1). Das Kapitel gliedert sich wie folgt: Die jeweiligen Kategorien verwandter Ansätze werden in den Abschnitten 4.1-4.6 im Detail diskutiert. Abschnitt 4.7 bewertet diese verwandten Ansätze auf Grundlage der im Kapitel 2 ermittelten Anforderungen. Abschnitt 4.8 fasst das Kapitel zusammen.

4.1 Forschungsprojekte

Es werden zunächst einschlägige Forschungsprojekte diskutiert, die im Kontext dieser Arbeit relevant sind.

Ziel des *DEMAC*-Projekts [DEM14] ist es: "als Weiterentwicklung der verteilten Prozessausführung auf Basis dienstorientierter Architekturen, Vorschläge zu erarbeiten, um die Entscheidung über die Verteilung eines bestehenden fachlichen Prozesses sowie dessen Ausführung, Interaktion mit menschlichen Akteuren, Überwachung und ggf. dessen Anpassung zur Laufzeit des Prozesses zu ermöglichen." Vor allem zwei Aspekte stechen heraus: Erstens werden verteilte Prozesse betrachtet. Zweitens erfolgt diese Behandlung auf Basis dienstorientierter Architekturen. Die Verteilung wird auf Grundlage der Prozessausführungssprache WS-BPEL [BPE14a] realisiert. Technische Bausteine sind dementsprechend Web Services und deren verteilter Ausführung, die auch auf mobilen Systemen erfolgen kann. Obwohl es im vorliegenden Ansatz nicht explizit um die verteilte Ausführung von Prozessen geht, sind einige der in *DEMAC* behandelten Aspekte hilfreich.

In [ZHKL10] werden Mechanismen für die Zuordnung von Aktivitäten zu Ausführungseinheiten vorgestellt. Diese basieren auf den drei Fähigkeiten (1) Algorithmen zu definieren, (2) Qualitätsmerkmale zu definieren und den (3) Ausführungskontext auszuwerten. *DEMAC* sieht diese Möglichkeiten vor, überlässt es aber dem konkreten Szenario, diese individuell zu realisieren.

	Bezeichnung	Beschreibung	Arbeiten
K1	Forschungsprojekte	Es gibt eine Reihe von Forschungsprojekten, die sich mit mobilen Prozessen beschäftigen.	<i>DEMAC</i> [DEM14, Zap05], <i>SLIVER</i> [HHGR06], <i>ROME4U</i> [ROM14], <i>SAMPROC</i> [SAM14], <i>CiAN</i> [SRG08]
K2	Ausgewählte Forschungsansätze	Es konnten ausgewählte Forschungsansätze identifiziert werden, die zu berücksichtigen sind.	[AGA ⁺ 95], [HS09], [KG04b, KG04a], [PT06], [KOK ⁺ 11, KOKC12, KOKC13], [JHS ⁺ 99], [KB09], [WCK11], [BTR10], [RS03], [Ham09], [BPE14b, RA07], [Haa13], [Kre14], [HC04]
K3	Zuteilung von Endanwendern zu Aktivitäten	Da die Zuteilung mobiler Endanwender zu mobilen Aktivitäten ein zentraler Aspekt dieser Arbeit ist, sind verwandte Ansätze in dieser Kategorie besonders relevant.	[BFA97], [SCFY96], [SM02], [WC02], [RR05], [CK08b], [CRRC12b, RORC13, CRRC12a, CGR ⁺ 13, CRRC11a, CKR ⁺ 15, CRRC13, CRORC12], [Mue04], [LCH ⁺ 11], [BPE14b, RA07]
K4	Prozess-Constraints	Da Prozess-Constraints mitbetrachtet werden, sind auch verwandte Ansätze zu dieser Kategorie identifiziert worden.	[BFA99], [Cra03], [Cra04], [PMR13], [PMR14], [Ly13, LRMGD09], [CK08a]
K5	Mobiler Kontext	In dieser Kategorie werden verwandte Ansätze betrachtet, die den mobilen Kontext betrachten.	[UELW10], [CEM01b], [PÁFMO09], [RS03], [YLD ⁺ 12], [LKSP02], [KZL07a], [KOK ⁺ 11, KOKC12, KOKC13]
K6	Existierende Technologien	Existierende Technologien und Infrastrukturen sind ebenfalls verwandte Ansätze.	<i>SAP Mobile Infrastructure</i> [SAP14d, SAP14a], <i>IBM Process Manager</i> [IBM14c]

Tabelle 4.1: Kategorien verwandter Ansätze

Was den Kontextaspekt bei der Zuteilung von Aktivitäten zu Ausführungseinheiten angeht, wird ein generischer Ansatz gewählt [KZL07a]. Konkret werden abstrakte Dienstklassen eingeführt, um zur Laufzeit den am besten geeigneten Web Service auszuwählen. Allerdings handelt es sich hierbei um automatische Aktivitäten, wohingegen die vorliegende Arbeit interaktive Aktivitäten betrachtet. Des Weiteren wurde im *DEMAC*-Projekt eine Prozessausführungssprache entwickelt [KZL06]. Es wird gezeigt, inwieweit deren Konzepte auf andere Prozessausführungssprachen übertragbar sind. Ferner werden in [ZHKL10] Sicherheitsaspekte betrachtet, die in der vorliegenden Arbeit jedoch ausklammert werden [ZHKL10]. Da in *DEMAC* Web Services betrachtet werden, spielt die Sicherheit eine andere Rolle als in der vorliegenden Arbeit. Web Services kommunizieren meist über unsichere Verbindungen (z.B. HTTP). Insgesamt werden die in dieser Arbeit betrachteten Konzepte im *DEMAC*-Metamodell berücksichtigt, spielen dort aber infolge der Fokussierung einer verteilten Prozessausführung nur eine untergeordnete Rolle.

Im *SLIVER*-Projekt wurde eine WS-BPEL Engine entwickelt, mit der sich auf Smart-Mobilgeräten Prozesse in autonomer Art und Weise ausführen lassen. Ziel des Projekt ist

es, trotz beschränkter Ressourcen eines Smart-Mobilgeräts, komplexe Prozesse ausführen zu können. Herausforderungen ergeben sich aber nicht nur aufgrund der beschränkten Ressourcen des Smart-Mobilgeräts, sondern es muss weiters berücksichtigt werden, dass ein Smart-Mobilgerät spontan wechselnde Adressen haben kann (z.B. beim Netzwechsel). Da WS-BPEL Prozesse externe Ressourcen rufen, muss mit solchen spontanen Adresswechseln umgegangen werden können. *SLIVER* zeigt, welche Architekturbestandteile im Kontext mobiler Prozesse wichtig sind und inwieweit Protokolle dort eine Rolle spielen. Enge verwandte Aspekte zur vorliegenden Arbeit konnten nicht identifiziert werden. Nichtsdestotrotz realisiert *SLIVER* bislang die komplexeste Ausführung mobiler Aktivitäten auf einem Smart-Mobilgerät.

Im *ROME4U*-Projekt wird, ähnlich wie in *SLIVER*, eine Engine entwickelt, die WS-BPEL Prozesse autonom auf Smart-Mobilgeräten ausführen kann. Was heraussticht und im Kontext der vorliegenden Arbeit relevant erscheint, ist dass *ROME4U* die Zuteilung mobiler Aktivitäten zu mobilen Endanwendern als wichtig einstuft. Daher werden die mobile Prozess-Engine und die technische Komponente, welche die Aktivitäten zuordnet, voneinander abgekoppelt. So kann der Aktivitäten-Handler auf demjenigen Smart-Mobilgerät ausgeführt werden, dessen mobiler Endanwender die am besten geeignete Ressourcensituation vorweist. In einem Katastrophenszenario könnte das der Katastrophenleiter sein, der die Aktivitäten dann seinen Mitarbeitern zuweist. Weiters wird in *ROME4U* der Kontext berücksichtigt, etwa ob die Batterieladestatus unter 20 Prozent fällt und dieses *Ereignis* berücksichtigt werden soll [BGF⁺09]. Jedoch stellen solche *Ereignisse* in *ROME4U* Ausnahmen dar und werden nicht, wie im vorliegenden Ansatz, bereits bei der Zuteilung berücksichtigt. Konkrete Strategien für die Ausnahmebehandlung werden ebenfalls nicht diskutiert. Dennoch zeigt *ROME4U*, dass eine geeignete Zuteilung mobiler Endanwender zu mobilen Aktivitäten besonders wichtig ist. Ebenso wie *SLIVER* ist *ROME4U* auf WS-BPEL als Ausführungssprache beschränkt.

Im *SAMPROC*-Projekt wird ebenfalls WS-BPEL verwendet, wenn auch aus einer anderen Perspektive. Es wird über die Prozessbeschreibung festgelegt, wie Code-Migration auf dynamische Weise mittels Peer-to-Peer-Protokoll durchgeführt werden kann. Kontextbezogen kann zur Laufzeit der komplette Ausführungszustand eines auf dem Smart-Mobilgerät laufenden Programms auf ein anderes Smart-Mobilgerät übertragen werden. Im vorliegenden Ansatz wird davon ausgegangen, dass nicht der gesamte Code migriert wird, sondern nur relevante Prozessdaten ausgetauscht werden: Ein Smart-Mobilgerät besitzt einen Prozess-Klienten, der mit einem *PrMS* kommuniziert und Standardaufgaben übernehmen kann. Diese Arbeit wird zeigen, dass dieser Prozess-Klient auch mit weiteren Anwendungsprogrammen auf dem Smart-Mobilgerät kommuniziert. Soll das ausführende Smart-Mobilgerät einer mobilen Aktivität ohne Zuhilfenahme des *PrMS* auf ein anderes Smart-Mobilgerät migriert werden, ist *SAMPROC* dazu in der Lage. Diese Arbeit betrachtet diesen Aspekt nicht.

Im Gegensatz zu den bisher vorgestellten Projekten wird in *CiAN* ein Choreographie-Modell anstelle eines Orchestrierungs-Ansatzes¹ betrachtet (vgl. Kapitel 3). Über ein neu entwickeltes Protokoll werden Prozessausführungszustände zwischen Smart-Mobilgeräten ausgetauscht. Bei der Zuteilung und Weitergabe werden einerseits der Kontext und andererseits Ausnahmesituationen berücksichtigt. Wie für eine Choreographie erläutert, wird in *CiAN* in einer Planungsphase zu Beginn der Prozessausführung festgelegt, wer welche Aktivität ausführen soll. Obwohl ausfallende Smart-Mobilgeräte berücksichtigt werden und die Austauschprotokolle sehr fortschrittlich sind, muss zu Beginn der Ausführung festgelegt werden, wer welche Aktivität ausführt. Des Weiteren werden mobile Endanwender nicht von automatischen Aktivitäten

¹ siehe [RS04] für eine Abgrenzung von Choreographie und Orchestrierung

unterschieden, d.h. es werden implizit nur automatische Aktivitäten berücksichtigt. Dennoch zeigt auch dieser Ansatz, dass Protokollbetrachtungen im Kontext von Smart-Mobilgeräten als Ausführungseinheiten für Prozesse wichtig sind.

An den vorgestellten Projekten fällt auf, dass bis auf *CiAN* alle WS-BPEL als Grundlage verwenden. Dass dies nicht ausreicht, wird in [GCFP10] ausführlich diskutiert. Dennoch wird in den jeweiligen WS-BPEL Ansätzen deutlich, dass die Protokolle zur Kommunikation mit den Smart-Mobilgeräten grundlegend sind. Ferner sticht ins Auge, dass die verteilte Ausführung bei allen Ansätzen eine wichtige Rolle spielt. Eine spezielle Behandlung einzelner mobiler Aktivitäten und deren Ausführungskontext nehmen diese Ansätze allerdings nicht vor. Einzig *SAMPROC* und *ROME4U* reißen diesen Aspekt an, wenn auch aus anderen Perspektiven.

4.2 Ausgewählte Forschungsansätze

In diesem Abschnitt werden ausgewählte Forschungsarbeiten diskutiert, die einzelne Aspekte betrachten, die auch im Kontext der vorliegenden Arbeit relevant sind.

[PT06] bietet einen Überblick zu Aspekten, die im Kontext mobiler Prozesse besonders wichtig sind. Weiters diskutiert [JHS⁺99], welche Anwendungsdomänen davon profitieren können, wenn Smart-Mobilgeräte in eine Prozessausführung integriert werden. In [CEM01a] wiederum wird aufgezeigt, dass mobile Informationssysteme komplex zu realisieren sind. Weiters wird diskutiert, welche Aspekte besonders zu berücksichtigen sind. Eine Kernaussage lautet, dass eine Realisierung stets in Richtung der *Awareness* für mobile Endanwender gehen sollte. Das heißt, es muss abgewägt werden, welche Aspekte dem mobilen Endanwender explizit vor Augen geführt werden sollen (z.B. Batterieladestatus) und welche nicht (z.B. Ort, an dem sich andere Akteure befinden). Eine solche *Awareness* muss für den Endanwender verständlich bleiben, da andernfalls keine Akzeptanz gegeben ist.

In [KG04b, KG04a] werden fachliche Prozesse untersucht, ob sich Teile davon eignen, mobil ausgeführt zu werden. Ferner wird behandelt, wie diese Teilprozesse mobil ausgeführt werden können, ohne den Prozess strukturell zu verändern. Da hier keine Unterscheidung zwischen automatischen und interaktiven Aktivitäten gemacht wird und eine verteilte Prozessausführung kein Ziel darstellt, sind [KG04b, KG04a] ausschließlich aus fachlicher Sicht zur Identifikation mobiler Aktivitäten relevant. Auch wenn die Identifikation mobiler Teilprozesse nicht im Fokus dieser Arbeit steht, sind entsprechende Betrachtungen aus fachlicher Sicht wichtig.

In [BTR10] wird für die Medizindomäne untersucht, inwieweit bestimmte Teile von Prozessen mobil ausgeführt werden können. Diese Arbeit umfasst eine sehr umfangreiche fachliche Analyse, die in anderen Arbeiten fehlt. Daraus lassen sich wiederum die Parameter dieser Arbeit gut als Anforderung ablesen.

In [KB09] wird für verteilte mobile Prozesse untersucht, inwieweit Daten, die durch den Prozess erzeugt werden, als Kriterium für eine geeignete Verteilung dienen können. Interessant an dieser Arbeit ist die explizite Betrachtung von Datenelementen im Kontext mobiler Prozesse. [Haa13] stellt die notwendigen Bausteine einer Infrastruktur vor, um Teile eines Prozesses an Smart-Mobilgeräte zuzuteilen. Dies hat auch die in der vorliegenden Arbeit vorgestellte Architektur inspiriert. In [Kre14] wird eine grundlegende Betrachtung zu leistungsfähigen Infrastrukturen für *PrMS* gegeben. In Anlehnung an diese Konzepte wird in der vorliegenden Arbeit eine geeignete Infrastruktur zur robusten Ausführung mobiler Aktivitäten konzipiert und realisiert. Einen wichtigen Beitrag bietet [AGA⁺95]. Diese Arbeit stellt ein Protokoll für die Realisie-

rung von Offline-Aktivitäten vor. Das in dieser Arbeit verwendete Protokoll (vgl. Kapitel 8) für Offline-Aktivitäten basiert teilweise auf diesem Ansatz, insbesondere um die Ausnahmebehandlung zu realisieren. Das Protokoll musste allerdings erweitert werden, um den mobilen Kontext geeignet einzubeziehen.

Der in [HS09] vorgestellte Ansatz zeigt, wie die Ausnahmebehandlung mobiler Aktivitäten mittels Transaktionen realisiert werden kann. Insbesondere werden Konzepte vorgestellt, um Ausfälle mobiler Aktivitäten zu kompensieren. Dies inspirierte auch das in Kapitel 9 vorgestellte Delegationskonzept. Außerdem werden sog. Properties für mobile Aktivitäten definiert, was vereinfacht ausgedrückt den in dieser Arbeit vorgestellten Parametern für den Kontext einer mobilen Aktivität entspricht. In [HS09] werden allerdings nur wenige Parameter definiert, was diesen Ansatz einschränkt.

[KB09] diskutiert ein generisches Konzept, um benutzerfreundliche mobile Anwendungen zu entwickeln. Dieses Konzept inspirierte unsere Ausnahmebehandlung für mobile Aktivitäten und sensibilisierte uns dafür, dass die Präsentation für den Endanwender auf dem Smart-Mobilgerät besonders wichtig ist. Weiters wird in [HC04] ein Konzept vorgestellt, um in einer Service-orientierten Umgebung auftretende Ausnahmen zu behandeln. Im engeren Sinne konnten keine Überschneidungen gefunden werden, dennoch werden in dieser Arbeit die möglichen Herausforderungen sehr gut diskutiert.

In [Ham09] wird diskutiert, wie parallele Zweige eines Prozesses, die von verschiedenen Smart-Mobilgeräten ausgeführt werden, synchronisiert werden sollen. Das dazu verwendete Protokoll ist für unsere Arbeit relevant und hat uns als Konzept zur Ausnahmebehandlung inspiriert.

Dieser Abschnitt hat ausgewählte verwandte Ansätze vorgestellt, die im Kontext dieser Arbeit relevant sind. Einige der Aspekte wurde übernommen und erweitert bzw. für unseren Einsatzzweck angepasst. Keine der diskutierten Arbeiten verfolgt den in dieser Arbeit vorgestellten Ansatz.

4.3 Zuteilung von Endanwendern zu Aktivitäten

Unserem Wissens nach gibt es für die Endanwenderzuteilung keine Ansätze, welche diese explizit und kontextbezogen für mobile Aktivitäten betrachten. Nichtsdestotrotz sind in Verbindung mit Endanwenderzuteilungen drei Kategorien verwandter Ansätze relevant.

Es gibt eine Reihe verschiedener Ansätze in der Literatur, um Endanwendern durch ein *PrMS* die für sie relevanten Aktivitäten zuzuteilen. Dies wiederum hat Auswirkungen auf die Generizität eines Ansatzes. In [Cra04] werden die grundsätzlichen Möglichkeiten diskutiert und verglichen. Im Kontext des Zuteilungsschemas muss auch BPEL4PEOPLE [BPE14b, RA07] betrachtet werden. Dieser Ansatz erweitert WS-BPEL um das Konzept interaktiver Aktivitäten. BPEL4PEOPLE wiederum basiert auf WS-HumanTask [WSH14]. Sowohl BPEL4PEOPLE als auch WS-HumanTask sind, im Gegensatz zu WS-BPEL, keine Standards. In BPEL4PEOPLE werden verschiedene Zuteilungsschemas für interaktive Aktivitäten vorgestellt. Diese werden in der vorliegenden Arbeit vor dem Hintergrund des Kommunikationsprotokolls zwischen *PrMS* und Prozess-Klient untersucht (vgl. Kapitel 8), wobei wichtige Aspekte erweitert wurden.

Die Varianten, ein Organisationsmodell zu realisieren, sind ebenfalls vielfältig. Daher wurden verwandte Ansätze dazu ebenfalls identifiziert [BFA97, SCFY96, SM02, WC02, RR05, CK08b, Mue04, LCH⁺11, BPE14b]. Auf dieser Basis wurde das Organisationsmodell dieser Arbeit abgeleitet.

Bei der Endanwenderzuteilung wird der mobile Kontext genutzt, um dynamisch zu entscheiden, welcher mobile Endanwender am besten geeignet für die Ausführung ist. Die Literatur spricht in diesem Kontext von dynamischer Ressourcenallokation. Allerdings wird diese bislang nur im nicht-mobilen Fall in folgenden Arbeiten behandelt: [CRRC12b, RORC13, CRRC12a, CGR⁺13, LRW09c]. Diese Ansätze wurden für diese Arbeit explizit berücksichtigt, jedoch unterscheiden sie sich erheblich von unserem Ansatz (vgl. Kapitel 8).

4.4 Prozess-Constraints

Im Kontext mobiler Prozessausführung sind uns keine verwandten Ansätze bekannt, die Prozess-Constraints (z.B. Separation of Duties) explizit betrachten. Im Falle einer nicht-mobilen, aber verteilten Ausführung von Prozessen, können Ansätze in [Bau01] gefunden werden. Was für diese Arbeit wichtig ist, sind die in Verbindung mit Prozess-Constraints zu betrachteten Beschränkungen. Um diese angemessen zu berücksichtigen, sind verschiedene Grundlagenarbeiten relevant: [BFA99, Cra04, Cra03, CK08a]. In diesen Arbeiten werden vor allem zwei Aspekte diskutiert: Erstens wird betrachtet, welche Prozess-Constraints generell relevant sind und wie diese kombiniert werden können. Wenn Prozess-Constraints zum Einsatz kommen, muss zweitens der Aspekt der korrekten Ausführung betrachtet werden. Zu beiden Aspekten stellt diese Arbeit eine konkrete Lösung für die Ausführung constraint-behafteter mobiler Aktivitäten vor.

4.5 Mobiler Kontext

Die Anwendung des mobilen Kontexts bei der Ausführung mobiler Aktivitäten wird in in verwandten Ansätzen kaum diskutiert. Allerdings gibt es verwandte Ansätze zur Erfassung des mobilen Kontexts. Die Möglichkeiten, einen Kontext zu erfassen, sind vielfältig [DAS01, MMY05]. Im Zusammenhang mit mobilen Aktivitäten gilt es vor allem, Parameter zu finden, welche ihre Zuteilung zu mobilen Endanwendern geeignet unterstützen. Diese werden in den später vorgestellten Algorithmen ausgewertet, um eine automatische Zuteilung mobiler Aktivitäten kontextbezogen zu realisieren.

Die in [RS03] verwendeten Parameter für mobile Aktivitäten haben die Überlegungen in dieser Arbeit inspiriert. Insbesondere werden die identifizierten Parameter des Smart-Mobilgeräts und des verwendeten Netzes zur Online-Verbindung übernommen. Allerdings wird in [RS03] nur eine kleine Teilmenge der in dieser Arbeit unterstützten Parameter behandelt. Einen weiteren Ansatz, der explizit Parameter erfasst, wird in [CEM01b] beschrieben. Hier wird besonders der Netztyp einer Online-Verbindung diskutiert. Die Verbindungstypen wurden aus [CEM01b] übernommen und erweitert (vgl. Kapitel 7).

In [YLD⁺12] wird für den mobilen Kontext ein expliziter Zusammenhang zwischen der Lokation der auszuführenden Tätigkeit und der Lokation des Endanwenders hergestellt. Dazu werden sog. Endanwenderprofile eingeführt. Ein solches wird vom Endanwender selbst definiert. Es gibt beispielsweise an, an welchen Orten er sich bevorzugt aufhält oder welche Daten ihn besonders interessieren. Auf Basis des Profils wird insbesondere die Datenübertragung zum Smart-Mobilgerät des Endanwenders berücksichtigt. [YLD⁺12] betrachtet allerdings keine Prozessumgebung. Dennoch inspirierte uns diese Arbeit, insbesondere das hier verfolgte Konzept

zum Abgleich der Lokation von Endanwendern und derjenigen an dem eine Aktivität auszuführen ist.

In [KOK⁺11, KOKC12, KOKC13] wird ein Kontextmodell für mobile Prozesse vorgestellt. Jedoch wird der Kontext für eine verteilte Prozessausführung eingesetzt; er beschreibt daher nicht einzelne Parameter mobiler Aktivitäten. Dennoch wird für eine mobile Prozessausführung ein umfassender Kontextbegriff eingeführt, der allerdings einem anderen Zweck dient als er in dieser Arbeit verfolgt wird.

Der in Kapitel 7 eingeführte mobile Kontext ist im Vergleich zu den diskutierten Arbeiten kategorisiert. Darüber hinaus ist er wesentlich umfangreicher und ermöglicht somit eine kontextbezogene Ausführung mobiler Aktivitäten unter vielen Gesichtspunkten. Da sich Aspekte in einer mobilen Umgebung oftmals schnell ändern können, ist eine differenzierte Betrachtung des mobilen Kontexts wichtig, um eventuell Alternativen automatisch bewerten zu können.

4.6 Existierende Technologien

Professionelle Software zur Ausführung mobiler Prozesse bzw. Aktivitäten in einer mit unserem Ansatz vergleichbaren Mächtigkeit gibt es aktuell nicht. Die fortschrittlichsten Plattformen, die es zu nennen gilt, bieten *SAP* und *IBM*.

In [Use13] haben wir auf Basis der *SAP Mobile Plattform* eine Anwendung realisiert, die mit Smart-Mobilgeräten die Wartung eines Schienennetzes für ein Bahnunternehmen ermöglicht. Insbesondere haben wir untersucht, welche Möglichkeiten existieren, um ein SAP-Backend anzusprechen. Abbildung 4.1 zeigt die grafische Notation mittels der Funktionsbausteine entwickelt werden. Letztere greifen auf ein SAP-Backend zu und können auf Smart-Mobilgeräte übertragen werden. Die Möglichkeiten, individuelle mobile Anwendungen zu entwickeln, sind umfangreich. Auch lassen sich Bausteine entwickeln, mit denen Smart-Mobilgeräte an Prozessen partizipieren können. Es handelt sich um individuelle mobile Anwendungen, die dem Prinzip der hybriden, mobilen Anwendung folgen (vgl. Kapitel 3). Ein generisches Prinzip, mobile Prozessaktivitäten unter Berücksichtigung des Kontexts zu realisieren, wird bei SAP nicht verfolgt. Vielmehr muss für jeden Anwendungsfall eine neue mobile Anwendung realisiert werden (vgl. Abb. 4.1). Mobile Prozessaktivitäten entsprechen in SAP daher jeweils individuellen Entwicklungen.

IBM folgt einem ähnlichen Prinzip. [IBM14c] beschreibt, wie mobile, interaktive Aktivitäten ausgeführt werden. Dazu muss im Prozessmodell analog zu unserem Ansatz die Aktivität als mobil markiert werden (vgl. Abb. 4.2). Danach können Smart-Mobilgeräte diese Aktivitäten als mobile Web-Anwendung ausführen.

Eine kontextbezogene Zuteilung mobiler Aktivitäten oder die Behandlung von Ausnahmen werden allerdings weder von *IBM* noch von *SAP* verfolgt. Ebenso wenig werden Prozess-Constraints unterstützt. Sowohl SAP als auch IBM bieten Entwicklungsumgebungen an, um mobile Anwendungen zu realisieren, die in unserer Terminologie einzelnen mobilen Aktivitäten entsprechen. Eine kontextbezogene Zuteilung von Aktivitäten, eine Ausnahmebehandlung oder gar eine Realisierung mobiler Aktivitäten werden weder von SAP noch IBM betrachtet.

Ein Aspekt, der bisher von verwandten Ansätzen kaum berücksichtigt wird und der auch in dieser Arbeit nicht weiter verfolgt wird, ist eine geeignete Modellierungsumgebung für mobile Prozesse und mobile Prozessaktivitäten. Im *MARPLE*-Projekt [MAR14] haben wir allerdings für Prozess-Fragmente, die auf Smart-Mobilgeräte übertragen werden, eine solche Modellierungs-

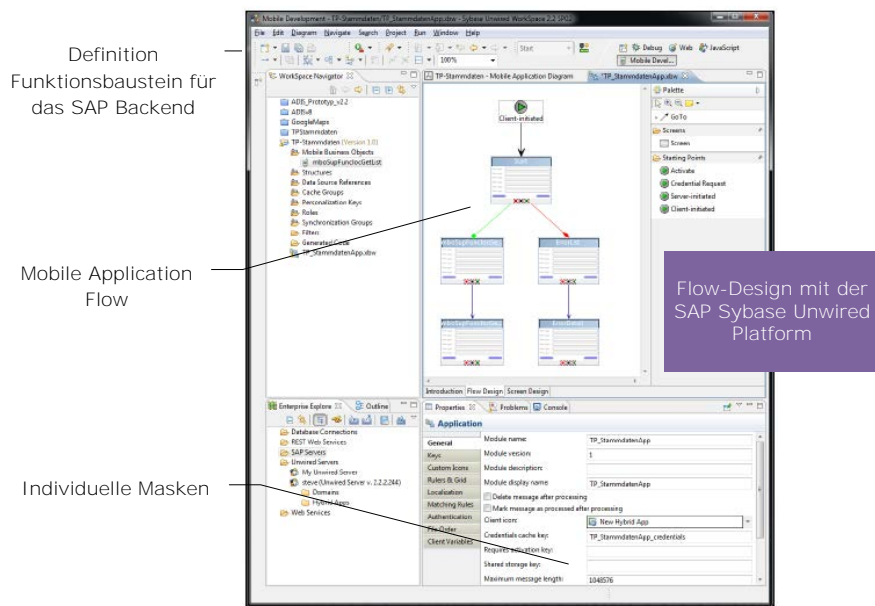


Abbildung 4.1: Flow-Design mit der SAP Sybase Unwired Platform [Use13]

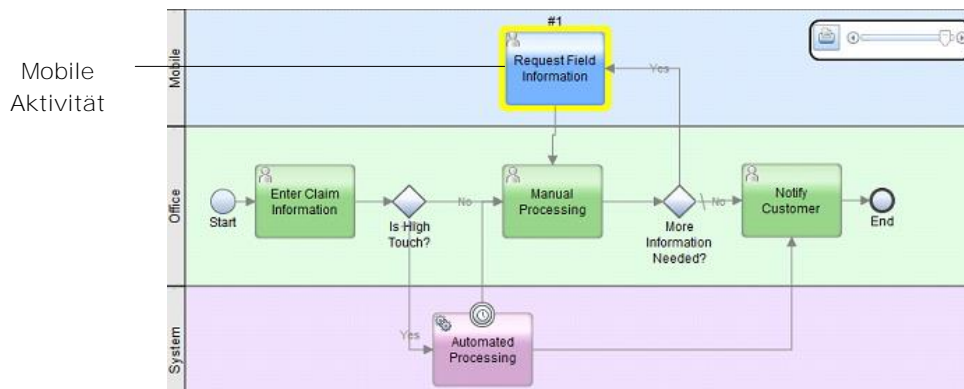


Abbildung 4.2: IBM: Markieren einer mobilen Aktivität [IBM14c]

umgebung prototypisch realisiert (vgl. Abb. 4.3) [PTKR10]. Da diese Arbeit auf die Ausführung mobiler Aktivitäten fokussiert, gehen wir hierauf aber nicht im Detail ein.

4.7 Diskussion

In diesem Abschnitt werden die in Kapitel 2 hergeleiteten Anforderungen herangezogen, um existierende Ansätze und damit den Stand der Technik zu bewerten. Tabelle 4.2 zeigt, dass viele der in dieser Arbeit adressierten Anforderungen in verwandten Ansätzen aktuell nicht adressiert werden. Insgesamt deckt der Stand der Technik viele der in dieser Arbeit adressierten Aspekte bislang nicht ab. Obwohl einige Ansätze die Wichtigkeit einer kontextbezogenen Zuteilung

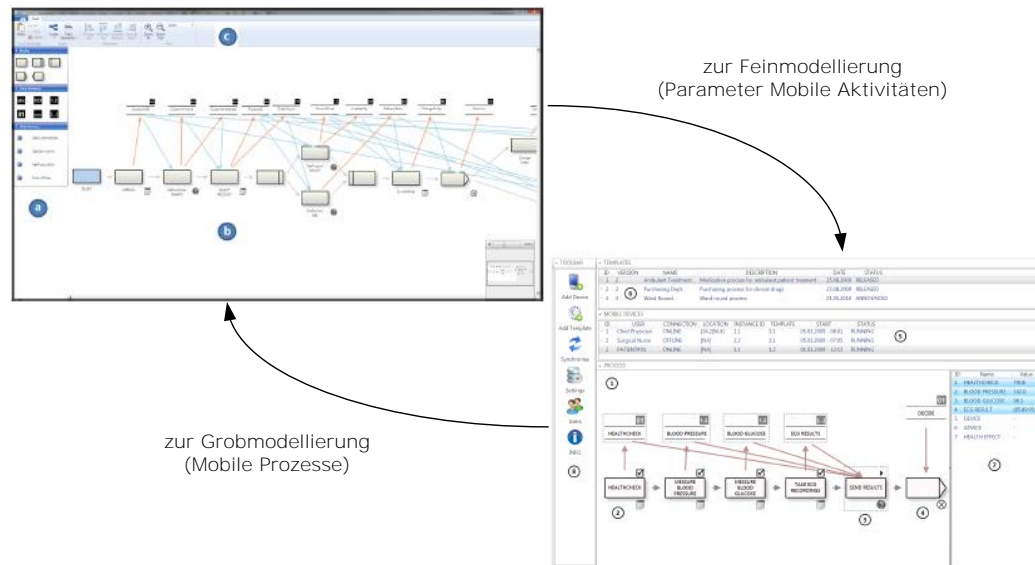


Abbildung 4.3: MARPLE Mediation Center

mobiler Aktivitäten betonen, gehen die hier vorgestellten Lösungen in Bezug auf die Ausführung mobiler Aktivitäten nicht weit genug.

Anforderung	Beschreibung	Stand der Technik
Funktionale Anforderungen mobiler Aktivitäten		
Anforderung 1	Kontextabhängige Zuteilung von Aktivitäten	X
Anforderung 2	Kontextabhängige Ausnahmebehandlung von Aktivitäten	X
Anforderung 3	Automatische Priorisierung von Aktivitäten	✓
Anforderung 4	Unterbrechen von Aktivitäten	✓
Anforderung 5	Übergabe von Aktivitäten	X
Anforderung 6	Systemwechsel für Aktivitäten	X
Anforderung 7	Spontanes Hinzufügen von Aktivitäten	✓
Anforderung 8	Berücksichtigung von Prozess-Constraints	X
Anforderung 9	Unterstützung von Offline-Aktivitäten	✓
Funktionale Anforderungen des mobilen Kontexts		
Anforderung 10	Berücksichtigung des Ausführungsorts	✓
Anforderung 11	Berücksichtigung des Netztyps	✓
Anforderung 12	Berücksichtigung des Formfaktors	X
Anforderung 13	Berücksichtigung des Energiestatus	✓
Anforderung 14	Berücksichtigung von Zeit-Constraints	✓
Anforderung 15	Berücksichtigung des Endanwenderverhaltens	X
Nicht-funktionale Anforderungen mobiler Aktivitäten		
Anforderung 16	Robuste Ausführung	X
Anforderung 17	Automatische Ausnahmebehandlung	X
Anforderung 18	Frühzeitige Erkennung von Laufzeitproblemen	X

Tabelle 4.2: Anforderungsübersicht dieser Arbeit und Stand der Technik

Anforderung	Bemerkung
Anforderung 2	Eine Ausnahmebehandlung mobiler Aktivitäten ist in einigen anderen Ansätzen zu finden, jedoch wird hier kein mobiler Kontext berücksichtigt.
Anforderung 3	Eine Priorisierung bei der Aktivitätenzuteilung wird bislang nur für Aktivitäten erforscht, die stationär ausgeführt werden. Eine Zuteilung mobiler Aktivitäten, unter Berücksichtigung des mobilen Kontexts, wird in der Literatur nicht diskutiert.
Anforderung 5, 6 und 8	Diese Anforderungen werden in verwandten Ansätzen zu mobilen Aktivitäten nicht berücksichtigt.
Anforderung 9	Offline-Aktivitäten werden für Prozesse diskutiert [AGA ⁺ 95], in denen keine mobilen Aktivitäten vorkommen. Jedoch beschreibt [AGA ⁺ 95] keinen Ansatz für mobile Aktivitäten. Ebenso wenig werden der mobile Kontext oder fortschrittliche Ausnahmebehandlungen behandelt.
Anforderungen 10-15	Obwohl einige der Parameter in verwandten Ansätzen vorkommen, betrachtet keine davon diese kombiniert. Ferner bezieht unser Ansatz essentielle Parameter ein, die in der Literatur bisher unberücksichtigt bleiben (z.B. Endanwenderverhalten).

Tabelle 4.3: Anmerkungen zu den Anforderungen

4.8 Zusammenfassung

In diesem Kapitel wurde der aktuelle Stand der Technik diskutiert, der für die vorliegende Arbeit relevant ist. Insbesondere wurde gezeigt, dass existierende Ansätze differenziert betrachtet werden müssen. Dazu wurden sie in verschiedene Kategorien eingeteilt und für jede Kategorie die jeweils relevanten Ansätze diskutiert. Im Anschluss wurde gezeigt, inwieweit die verwandten Ansätze, die in dieser Arbeit hergeleiteten Anforderungen adressieren. Dies bestätigte, dass sich die vorliegende Arbeit mit bisher nicht adressierten Herausforderungen (s. auch Tabelle 4.3) mobiler Aktivitäten befasst.

Teil II

Prozessumgebung

Männer sind zwar oft so jung, wie sie sich fühlen, aber niemals so bedeutend.

Simone de Beauvoir (1908-1986)

5

Prozess-Metamodell

In diesem Kapitel wird das Prozess-Metamodell vorgestellt, das dieser Arbeit zugrunde liegt. Die in den nachfolgenden Kapiteln ausgeführten Konzepte setzen auf diesem Metamodell auf. Allerdings ist das entwickelte Konzept zur robusten Ausführung mobiler Aktivitäten auch auf andere Metamodelle übertragbar. Das Prozess-Metamodell ist zudem ein wichtiger Bestandteil der in dieser Arbeit zugrunde gelegten Prozessumgebung.

Das Kapitel gliedert sich wie folgt: Abschnitt 5.1 leitet in das Thema ein und Abschnitt 5.2 behandelt den Lebenszyklus von Prozessen. In Abschnitt 5.3 wird das verwendete Prozess-Metamodell vorgestellt. Abschnitt 5.4 wiederum behandelt das Modell für die Zuteilung von Endanwendern zu Aktivitäten. Abschnitt 5.5 fasst schließlich das Kapitel zusammen.

5.1 Einleitung

Prozess-Metamodelle sind ein wichtiger Gegenstand aktueller Forschungsarbeiten [DLRMR13]. In der Praxis allerdings hat es lange gedauert, bis sich Standards etabliert hatten [Sto08]. Der derzeitige De-facto-Standard für die Modellierung von Prozessen ist die *Business Process Model and Notation (BPMN)* [BPM14a, BPM14b]. Nachteilig am BPMN-Metamodell war lange Zeit, dass es keine Ausführungsaspekte betrachtete. Vielmehr diente das verwendete Metamodell ausschließlich dazu, Geschäftsprozesse grafisch zu erfassen bzw. zu dokumentieren. Um die resultierenden Prozessmodelle ausführen zu können, verwendete man meist die Ausführungssprache WS-BPEL [BPE14a] aus dem Bereich Service-orientierter Architekturen. Die Transformation von *BPMN* zu *WS-BPEL* (und umgekehrt) ist wiederum ein aufwändiger und fehleranfälliger Prozess [Gao14, RM06, ODHA08].

Die aktuelle Spezifikation BPMN 2.0 zielt nun auch auf die Ausführung entsprechenden Prozessmodelle ab, d.h. eine operationale Semantik für BPMN 2.0 existiert [DVG10]. Ein Prozess-Metamodell sollte daher immer auch daran gemessen werden, ob es Ausführungsaspekte adressiert. Ebenso wichtig ist die korrekte Ausführbarkeit der spezifizierten Prozessmodelle. Damit wiederum gehen die Akzeptanz seitens der Anwender und die Einsatzmöglichkeiten in der Praxis einher. Diesen Umstand kann das Prozess-Metamodell unterstützen oder im ungünstigsten Fall komplexer gestalten. Bei einer Modellierung mit großen Freiheitsgraden, die eine automatische Überprüfung auf Korrektheit, etwa um Deadlocks zu vermeiden, nicht zulässt, muss dieser

Umstand eventuell später bei der Implementierung durch aufwändige Algorithmen kompensiert werden. Im ungünstigsten Fall können solche Ausnahmen gar nicht ausgeschlossen werden und eine Ausnahmebehandlung muss manuell zur Laufzeit erfolgen, was einen praktischen Ansatz stark einschränkt. Generell sind Prozess-Metamodelle zu bevorzugen, die eine korrekte Ausführbarkeit der beschreibbaren Prozessmodelle dediziert adressieren. Das hier vorgestellte Prozess-Metamodell folgt dem ADEPT-Ansatz [Rei00, RD97, RHD98]¹. Es hat einerseits den Vorteil, direkt ausgeführt werden zu können (für eine Beschreibung der operationalen Semantik siehe [Rei00, DRRM⁺09b, RDRM⁺09b]). Andererseits verfolgt es im Kern eine korrekte Modellierung und Ausführung, basierend auf dem sog. Correctness-by-Construction Prinzip [DRRM⁺09a]. Da in [Rei00] die formalen Definitionen dieses Prozess-Metamodells zu finden sind, beschränken sich die folgenden Ausführungen auf diejenigen Aspekte, die im Kontext dieser Arbeit relevant sind.

5.2 Prozess-Lebenszyklus

Abbildung 5.1 zeigt den generellen Prozess-Lebenszyklus mit seinen fünf typischen Phasen. Diese werden nachfolgend kurz vorgestellt. Prozesse werden zuerst in einer Modellierungsumgebung definiert. Dies geschieht üblicherweise in einer grafischen Notation. Modellierte Prozesse werden dann mithilfe der Modellierungsumgebung auf die Laufzeitumgebung eines *PrMS* übertragen (sog. Deployen von Prozessen). Das *PrMS* wiederum führt die Prozesse aus und ermöglicht außerdem deren Monitoring [Par10]. Die letzte Phase betrifft die Evolution von Prozessen. Wenn Prozesse die Modellierphase verlassen und ausgeführt werden, muss die Frage geklärt werden, wie man mit nachträglichen Änderungen (d.h. der Evolution von Prozessen) umgehen kann [RRD03]. Solche Änderungen stellen besonders dann eine Herausforderung dar, wenn Prozesse bzw. Prozessinstanzen langlaufend sind [RHD98]. Da sich bereits tausende Instanzen eines Prozesses in Ausführung befinden können, müssen entsprechende Änderungen am Prozess vor dem Hintergrund laufender Instanzen bewertet werden. Die Evolution von Prozessen ist daher intensiver Forschungsgegenstand [RHD98, Rin04, RMRW08].

In dieser Arbeit werden die Phasen der Modellierung, des Deployments und der Ausführung explizit adressiert. Das Monitoring und die Evolution von Prozessen liegen dagegen nicht im Fokus der vorliegenden Dissertation.

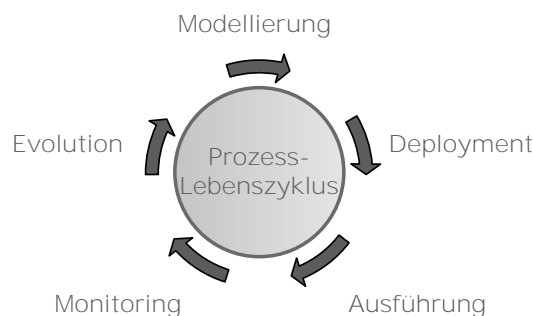


Abbildung 5.1: Prozess-Lebenszyklus

¹Weitere Arbeiten zum Vertiefen des ADEPT-Ansatzes sind [RD98, RRMD03, RRKD05, RD09].

	Bezeichnung	Beschreibung
①	Starten von Aktivitäten	Das Konzept zum Starten einer Aktivität wird für mobile Aktivitäten erweitert.
②	Ausnahmebehandlung von Aktivitäten	Das Konzept zur Ausnahmebehandlung einer Aktivität wird für mobile Aktivitäten erweitert.
③	Aktivitätenstruktur	Es wird für mobile Aktivitäten primär der atomare Strukturtyp betrachtet. Die Konzepte lassen sich problemlos auf Aktivitäten, die Sub-Prozesse modellieren, übertragen.
④	Archivierung bzw. Log-Einträge	Da mobile Aktivitäten ein erweitertes Zustandsmodell fordern und ferner ein mobiler Kontext mit neuen Parametern für mobile Aktivitäten entwickelt wird, wird die Archivierung mobiler Aktivitäten (bzw. Log-Einträge der Archivierung) erweitert.
⑤	Aktivitätentyp	Es werden nur interaktive mobile Aktivitäten betrachtet, d.h. die von Endanwendern (d.h. Personen) mobil ausgeführten Aktivitäten. Dem gegenüber stehen automatische Aktivitäten, die von einem <i>PrMS</i> ohne Interaktionen mit Endanwendern ausgeführt werden. ²
⑥	Zuteilungsart	Es wird die automatische Zuteilungsart eines <i>PrMS</i> für mobile Aktivitäten erweitert. Eine manuelle Zuteilung mobiler Aktivitäten wird nicht betrachtet.
⑦	Autorisierung	Die Bewertung bei der Autorisierung für die Zuteilung mobiler Aktivitäten zu mobilen Endanwendern wird erweitert.
⑧	Constraints	Die Bewertung von Prozess-Constraints, die für die Zuteilung mobiler Aktivitäten zu mobilen Endanwendern relevant sind, wird für mobile Aktivitäten erweitert.

Tabelle 5.1: Prozess-Metamodell und mobile Aktivitäten

Da der Kontrollflussgraph gemäß ADEPT-Ansatz für die vorgestellten Konzepte relevant ist, wird dieser kurz eingeführt (für Details und Ergänzungen siehe [Rei00]). Ein ADEPT-Kontrollflussgraph, im Folgenden der Einfachheit halber Prozess-Modell (PM) genannt, wird durch ein Tupel $PM = (N, E, D, NT, ET, V_{OUT}, V_{IN}, DP, EC, Template)$ beschrieben (vgl. Abb. 5.3). Ein Prozess-Modell besteht im Kern aus einer Menge von Knoten N , den Aktivitäten, und Kanten E , welche die Knoten verbinden. Ein Knoten hat einen Typ $NT(n) \in \{STARTFLOW, ENDFLOW, ACTIVITY, NULL, STARTLOOP, ENDLOOP\}$. Für die Kanten wiederum gilt $E \in N \times N \times EdgeTypes$. Ferner haben Kanten einen Kantentyp $ET(e) \in \{CONTROL_E, LOOP_E, SYNC_E\}$. Weiters existieren sog. Datenelemente D , die von einem oder mehreren Knoten gelesen bzw. geschrieben werden können.

Jeder Knoten hat außerdem eine Eingangssemantik V_{IN} und eine Ausgangssemantik V_{OUT} . Diese geben an, welche Kanten eingehend wie bewertet sein müssen, um den Knoten zu aktivieren bzw. welche Kanten ausgehend wie markiert werden sollen, um nach Ende der

² Allerdings stellt die Betrachtung ausschließlich interaktiver Aktivitäten keine Einschränkung der vorgestellten Konzepte dar. Automatische Aktivitäten sind aus Sicht der Zuteilung für ein *PrMS* weniger komplex und die Konzepte interaktiver Aktivitäten daher problemlos auf automatische Aktivitäten übertragbar.

Aktivität im Kontrollfluss voranzuschreiten. Die möglichen Typen der Eingangssemantik V_{IN} sind $\{NONE, ONE_Of_ONE, ALL_Of_ALL, ONE_Of_ALL\}$. Entsprechend sind die möglichen Typen der Ausgangssemantik V_{OUT} definiert. Da die Eingangs- und Ausgangssemantik auch bedingte Verzweigungen erlauben (z.B. $ONE_Of_ALL^3$), werden sog. Entscheidungsparameter DP für die Auswertung der Bedingung zur Laufzeit eingeführt. Entscheidungsparameter werden wiederum mit Datenelementen verknüpft (vgl. Abb. 5.3). Ferner existiert ein Auswahlcode EC , der bei bedingten Verzweigungen die relevanten Kanten aktiviert.

$$PM = (N, E, D, NT, ET, V_{OUT}, V_{IN}, DP, EC, Template)$$

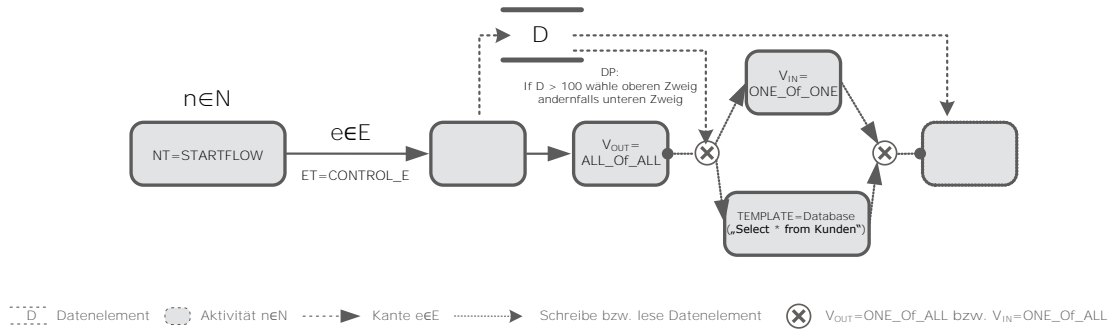


Abbildung 5.3: ADEPT-Prozessmodell (vereinfacht)

Schließlich ist jedem Knoten n des Prozessmodells ein *Template* zugeordnet. Es gibt an, welche Funktion die Aktivität ausführt (z.B. ein *Template* für einen Datenbankaufruf). Da sowohl für die Eingangssemantik V_{IN} als auch die Ausgangssemantik V_{OUT} die beiden Verzweigungsarten $XOR = ONE_Of_ALL$ und $AND = ALL_Of_ALL$ in Prozessbeispielen dieser Arbeit häufig verwendet werden, werden diese durch Abbildung 5.4 kurz illustriert.

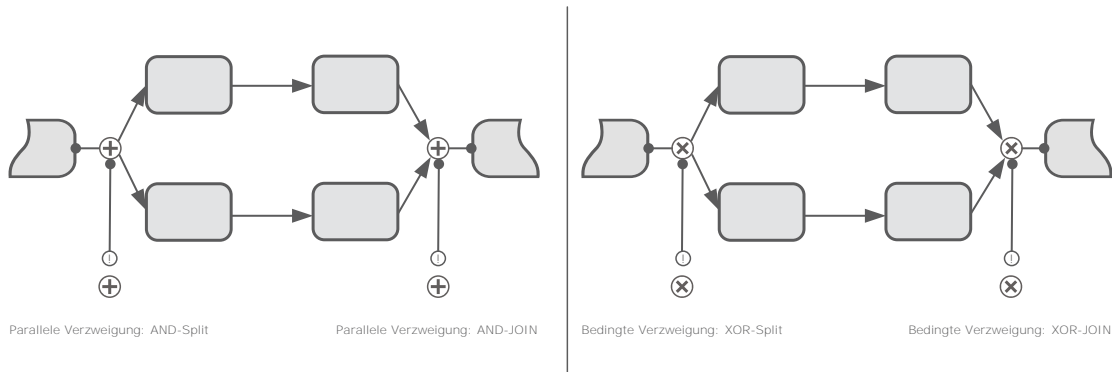


Abbildung 5.4: Darstellung bedingter Verzweigungen in Prozessen

Des Weiteren wird das Zustandsmodell für Aktivitäten und Kanten des Prozesses eingeführt. Diese Zustandsinformationen bilden die Voraussetzung für die Definition der operationalen Semantik des Metamodells [Rei00] und damit für die Ausführung von Prozessmodellen. Beide

³Auch als XOR-Verzweigung bezeichnet.

Zustandsinformationen sind für die vorliegende Arbeit von Bedeutung. Abbildung 5.5 zeigt die möglichen Zustände einer Aktivität und zulässige Zustandsübergänge.

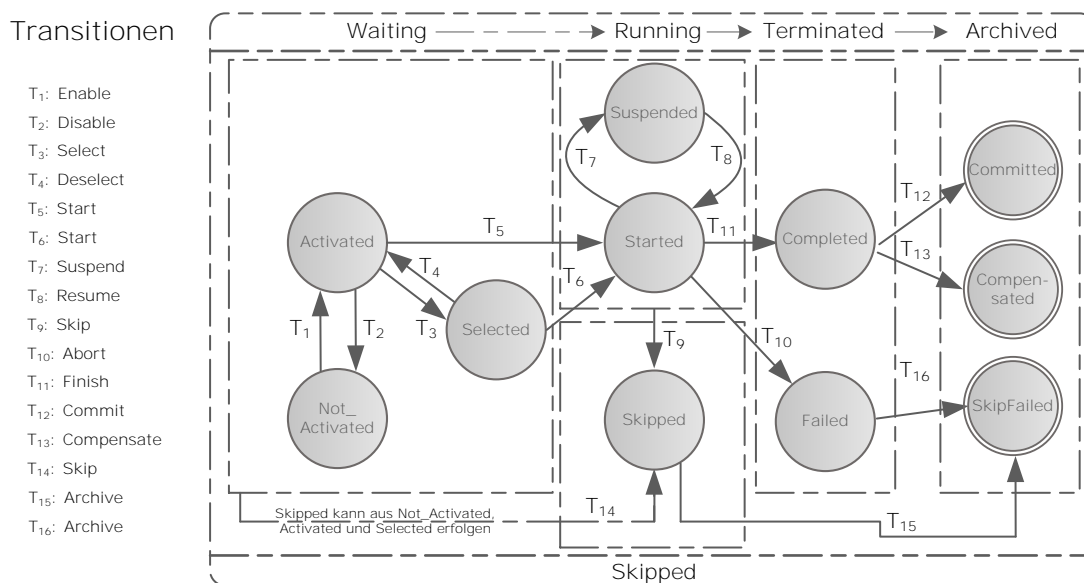


Abbildung 5.5: Statechart einer Aktivitäteninstanz (vereinfacht)

Der Zustand *Waiting* beschreibt eine Aktivität direkt nach Prozessstart. Wird die Ausführung der Aktivität gestartet, wechselt sie in den Zustand *Running*. Nach Beendigung der Ausführung wiederum resultiert der Zustand *Terminated*. Letzterer drückt aus, dass die Aktivität entweder erfolgreich beendet wurde (Sub-Zustand *Completed*) oder mit einer Ausnahme endete (Sub-Zustand *Failed*). Im Anschluss wechselt die Aktivität in den Zustand *Archived*, woraufhin die relevanten Ausführungsdaten in ein Log-File geschrieben werden.

In den Zuständen *Waiting* und *Running* kann die Aktivität ferner in den Zustand *Skipped* wechseln. In diesem Fall wird die Aktivität abgebrochen. Dieser Zustandswechsel hängt von weiteren Faktoren ab, da zum Beispiel nachfolgende Aktivitäten auf Daten angewiesen sein können, die durch *Skipped* der Aktivität nicht zur Verfügung stehen würden. D.h. ob die Aktivität den Zustand *Skipped* annehmen kann oder nicht, hängt von ihrer Einbettung in den Kontrollflussgraphen des Prozessmodells ab. Ferner kann *Skipped* für Aktivitäten notwendig werden, die nicht mehr zur Ausführung kommen (z.B. innerhalb eines Prozesszweiges, der wegen einer bedingten Verzweigung abgewählt wird).

Im Folgenden wird ein Aspekt des Zustandsmodells näher diskutiert, der im Kontext dieser Arbeit relevant ist. In Tabelle 5.1 wurde bei den Ausführungen zum Prozess-Metamodell bereits diskutiert, dass es zwei verschiedene Typen von Aktivitäten gibt: automatische und interaktive Aktivitäten. Erstere werden nicht von Personen ausgeführt, sondern von einem Rechner ohne Benutzerinteraktionen (z.B. Aufruf einer Datenbank oder eines Web Services). Interaktive Aktivitäten werden hingegen stets von Personen, im Folgenden Endanwender genannt, ausgeführt. Üblicherweise wird eine interaktive Aktivität einem Endanwender auf seinen stationären Rechner übertragen, der diese dann auf seinem Rechner bearbeitet und die eingegebenen Daten an das *PrMS* zurückschickt.

Die beiden Aktivitätentypen unterscheiden sich in Bezug auf das Durchlaufen der Aktivitäts-

tenzustände. Automatische Aktivitäten wechseln vom Zustand *Activated* stets in den Zustand *Started*, interaktive Aktivitäten gelangen dagegen in den Zustand *Started* nur indirekt über den dazwischen liegenden Zustand *Selected*. Letzterer wird aus zweierlei Gründen notwendig. Erstens kommen zumeist mehrere Endanwender infrage, um eine interaktive Aktivität zu bearbeiten. D.h. mehrere Endanwender qualifizieren sich für die Ausführung der Aktivität. In diesem Fall muss die Aktivität vor ihrer Bearbeitung zuerst von einem dieser Endanwender selektiert werden (was zum Wechsel in den Zustand *Selected* führt). Für alle anderen Endanwender, die ebenfalls qualifiziert sind, wird die Aktivität dann gesperrt. Zweitens muss derjenige, der sich für die Aktivität entscheidet, diese nicht sofort bearbeiten, sondern kann sie in seiner Arbeitsliste (sog. Worklists) verwalten und erst zu einem späteren Zeitpunkt ausführen.

Bei interaktiven Aktivitäten werden im Zustand *Activated* alle infrage kommenden Endanwender informiert. Sobald sich einer dazu entscheidet, die Aktivität auszuführen, wechselt die Aktivität in den Zustand *Selected*. Wählt dieser Endanwender die Aktivität auch tatsächlich in seiner Worklist aus, wechselt sie in den Zustand *Started*. **Die in dieser Arbeit betrachteten mobilen Aktivitäten sind interaktiv, d.h. wir betrachten nur solche Aktivitäten, die von einem Endanwender ausgeführt werden.**

Automatische Aktivitäten wechseln direkt vom Zustand *Activated* in den Zustand *Started*.

Abbildung 5.6 illustriert die Notationen für die verschiedenen Aktivitätszustände.



Abbildung 5.6: Darstellung von Aktivitätszuständen in Prozessen

Abschließend wird das Zustandsmodell für Aktivitäten ergänzt. Generell können Prozessaktivitäten mehrfach ausgeführt werden. Analog zu [Rei00] wird für Aktivitäten deshalb ein Iterationszähler eingeführt. Er gibt an, in welcher Ausführungsiteration sich eine Aktivität bzw. Aktivitäteninstanz befindet. Für eine Aktivität n gibt $It(n)$ an, in welcher Ausführungsiteration sich Aktivität n befindet.⁴ Generell wird die Instanz einer Aktivität n wie folgt identifiziert: $a = \langle n, it \rangle$ mit $n \in N$ und $it = t(n)$, $it \in \mathbb{N}_0$ gilt.

Abbildung 5.7 illustriert die verschiedenen Kantenzustände.



Abbildung 5.7: Darstellung von Kantenzuständen in Prozessen

Abbildung 5.8 veranschaulicht abschließend die Aktivitäten- und Kantenzustände an einem Prozessbeispiel.

⁴ $It(n) \in \{0, 1, 2, \dots\}$ ($It : N \mapsto \{0, 1, 2, \dots\}$)

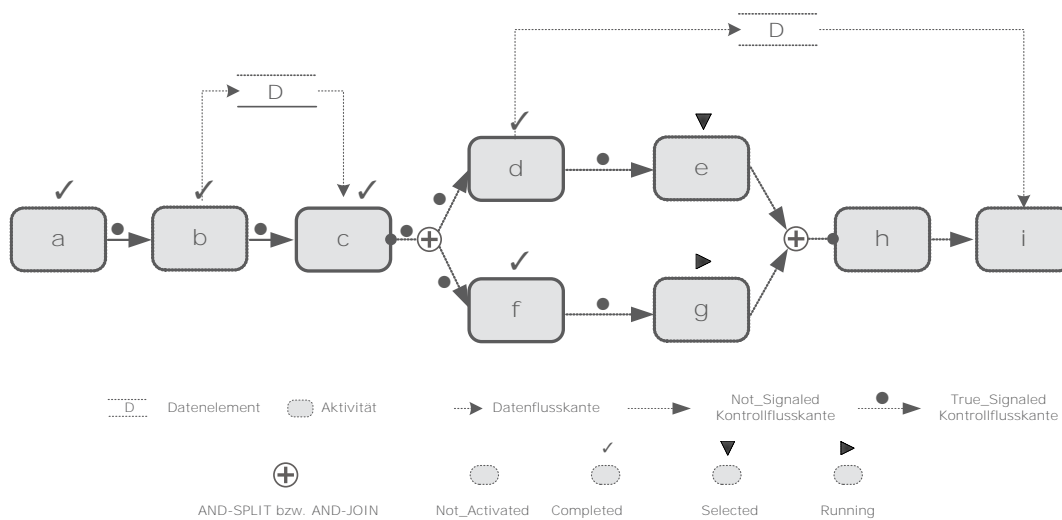


Abbildung 5.8: Prozessbeispiel mit Aktivitäten- und Kantenzuständen

Ein Prozessmodell umfasst ferner Datenelemente. Diese bilden die Grundlage für den Datenfluss, d.h. für das Lesen und Schreiben von Datenelementen [Rei00]. Dabei sind die verwalteten Datenelemente eines Prozesses global definiert. Bei der Modellierung eines Prozesses kann daher für jede Aktivität ein Lesezugriff und ein Schreibzugriff für jedes beliebige Datenelement definiert werden, sofern dies die Datenversorgung zulässt (d.h. ein Datenelement muss zuerst geschrieben werden, bevor es gelesen werden kann). Um Lost Updates bei konkurrierenden Zugriffen [BBG⁺95, GHOS96] (z.B. in parallelen Pfaden, die auf dasselbe Datenelement schreibend zugreifen) zu vermeiden, werden in [Rei00] entsprechende Mechanismen vorgestellt. Eine Illustration gibt Abbildung 5.9.

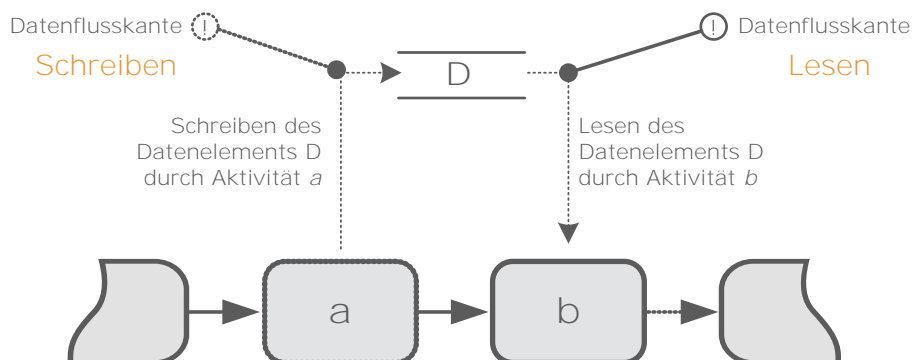


Abbildung 5.9: Darstellung des Datenflusses in Prozessen.

5.4 Endanwendermodell

Im Folgenden wird diskutiert, welche Varianten es für die Zuteilung interaktiver Aktivitäten an Endanwender gibt. Generell ist anzumerken, dass die Endanwenderzuteilung ein wichtiger Aspekt jedes *PrMS* ist. Da in dieser Arbeit die robuste Ausführung mobiler interaktiver Aktivitäten betrachtet wird, muss die Endanwenderzuteilung ebenfalls angemessen berücksichtigt werden. Nachfolgend wird diskutiert, welche Aspekte dabei eine Rolle spielen und welches Zuteilungsmodell zugrunde gelegt wird. Dazu werden die beiden Phasen der Modellierung und der Ausführung getrennt betrachtet.

5.4.1 Modellierzeit

Welche Endanwender sich zur Ausführungszeit für die Bearbeitung einer interaktiven Aktivität qualifizieren, wird in modernen *PrMS* meist zur Modellierzeit festgelegt. Varianten, die erst zur Ausführungszeit festgelegt werden, werden daher nicht weiter betrachtet. Auf Basis der Modellierung kann ein *PrMS* zur Ausführungszeit entscheiden, welche Endanwender konkret infrage kommen.

Die Literatur unterscheidet vielfältige Möglichkeiten zur Modellierung von Endanwenderzuordnungen [Rei00, Bau01, Rin04, RR05, WC02, SM02, BFA99, BFA97, CK08b, Cra03, SCFY96, PMR13]. In dieser Arbeit wird jeder interaktiven Aktivität n ein Endanwenderausdruck $EA(n)$ zugeordnet. Dieser gibt an, welche Endanwender sich zur Ausführungszeit für die Bearbeitung der interaktiven Aktivität n qualifizieren. Tabelle 5.2 stellt die möglichen Endanwenderausdrücke für $EA(n)$ vor. Es wird ferner unterschieden, ob es sich um constraint-behaftete Endanwenderausdrücke handelt oder nicht. Constraint-behaftete Endanwenderausdrücke spielen eine wichtige Rolle in dieser Arbeit, da sie für mobile Aktivitäten eine besondere Komplexität mit sich bringen. Diese sind aus praktischer Sicht sehr relevant, werden aber bislang im Kontext mobiler interaktiver Aktivitäten nicht betrachtet. Dass deren Unterstützung komplex ist, speziell vor dem Hintergrund der Robustheit, wird diese Arbeit zeigen und entsprechende Lösungen dazu vorstellen. In der Literatur können weitere Möglichkeiten für Endanwenderausdrücke als die hier vorgestellten gefunden werden. Da ein Hinzufügen weiterer Endanwenderausdrücke die Konzepte dieser Arbeit nicht beschränkt und die in Tabelle 5.2 vorgestellten Möglichkeiten eine komplexe Modellierung von Endanwendern zu interaktiven Aktivitäten ermöglichen, werden die Endanwenderausdrücke auf die in Tabelle 5.2 vorgestellten beschränkt. Diese werden schließlich vom *PrMS* ausgewertet, um zu entscheiden, welche Endanwender sich letztlich zur Ausführung einer Aktivität qualifizieren.

Eine Illustration constraint-behafteter Endanwenderausdrücke, im Folgenden Prozess-Constraints genannt, gibt Abbildung 5.10.

Zuletzt wird ein wichtiger Aspekt der Prozess-Constraints diskutiert. So gibt es Konstellationen, die nicht modelliert werden dürfen. Abbildung 5.11 zeigt eine nicht erlaubte Modellierung eines *Separation of Duties* Constraints. Aktivitäten d und g befinden sich in parallelen Prozesszweigen. Da für die Ausführung der beiden Aktivitäten keine zeitliche Reihenfolge existiert, etwa dass Aktivität d garantiert vor Aktivität g ausgeführt wird, kann dieser Prozess-Constraint zur Ausführungszeit nicht immer ausgewertet werden. Für die Prozess-Constraints EA3 und EA4 können solche Fälle auftreten. Daher können diese Constraints nur auf Aktivitäten angewandt werden, für die eine partielle Ordnung vorliegt (d.h. die zeitliche Reihenfolge der Ausführung ist

Bezeichnung	Beschreibung	Beispiel
Endanwenderausdrücke zur Autorisierung		
EA1 Verwendung von Organisationseinheiten	Endanwender können einer oder mehreren Organisationseinheiten (OE) angehören. Organisationseinheiten geben Bereiche eines Unternehmens an, denen der Endanwender zugehörig ist.	$EA(n) = OE(Notaufnahme)$
EA2 Verwendung von Rollen	Endanwender können einer oder mehreren <i>Rollen</i> angehören. Rollen realisieren Aufgabenbeschreibungen für Endanwender.	$EA(n) = Rolle(Arzt)$
Die Endanwenderausdrücke EA1 und EA2 können kombiniert werden.		$EA(n) = OE(Notaufnahme) \wedge Rolle(Arzt)$
Constraint-behaftete Endanwenderausdrücke		
EA3 Unterstützung von Binding of Duties	Dieser Endanwenderausdruck referenziert den Endanwenderausdruck einer anderen interaktiven Aktivität. Diese Referenz besagt, dass die beiden interaktiven Aktivitäten vom selben Endanwender ausgeführt werden müssen. Beispielsweise sollen die beiden interaktiven Aktivitäten <i>Blutdruck messen</i> und <i>Blut abnehmen</i> vom gleichen Arzt durchgeführt werden.	$EA(n) = EA(m)$
EA4 Unterstützung von Separation of Duties	Dieser Endanwenderausdruck referenziert den Endanwenderausdruck einer anderen interaktiven Aktivität. Die Referenz besagt, dass die beiden interaktiven Aktivitäten nicht vom selben Endanwender ausgeführt werden dürfen. Man spricht vom Vier-Augen-Prinzip. Beispielsweise sollen die beiden interaktiven Aktivitäten <i>Kredit Antrag ausfüllen</i> und <i>Kredit Antrag genehmigen</i> nicht vom gleichen Sachbearbeiter durchgeführt werden.	$EA(n) \neq EA(m)$
EA5 Unterstützung von Cardinality	Dieser Endanwenderausdruck besagt, dass eine interaktive Aktivitäten mehrfach vom selben Endanwender ausgeführt werden muss. Beispielsweise soll der <i>Blutdruck</i> mehrfach hintereinander bestimmt werden.	$EA(n) = EA(n)^z$
Die Endanwenderausdrücke EA1-EA5 können kombiniert werden.		$EA(n) = OE(Notaufnahme) \wedge EA(n) \neq EA(m)$

Tabelle 5.2: Verwendete Endanwenderausdrücke (vereinfacht)

eindeutig)⁵. In dieser Arbeit wird vorausgesetzt, dass die genannten Prozess-Constraints EA3 und EA4 nur auf Aktivitäten angewandt werden, für die eine partielle Ordnung gilt.

⁵siehe [CK08c, CK08a, Bau01, BD00b, BD00a] für Details

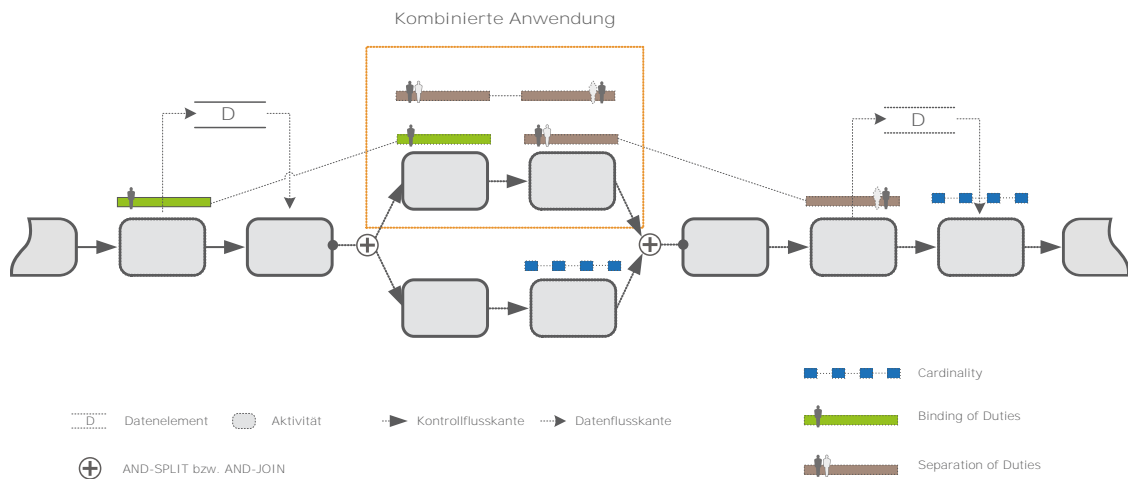


Abbildung 5.10: Darstellung von Prozess-Constraints

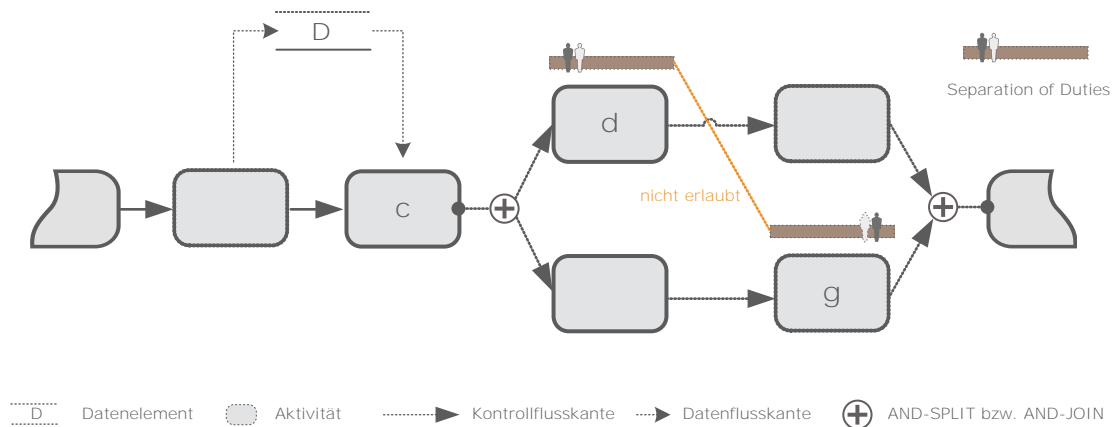


Abbildung 5.11: Nicht erlaubte Anwendung von Separation of Duties

5.4.2 Ausführungszeit

Für die Auswertung der vorgestellten Endanwenderausdrücke zur Ausführungszeit einer interaktiven Aktivität unterscheidet die Literatur mehrere Varianten [CK08c, CK08a, Bau01] (vgl. Tabelle 5.3).

Die statische Variante ist inflexibel. Sie wird daher in der Praxis kaum angewandt und in dieser Arbeit nicht weiter betrachtet. In dieser Arbeit wird die dynamische Variante II gewählt. Diese hat im Vergleich zur dynamischen Variante I den Nachteil, dass sich Änderungen an Endanwenderausdrücken zur Ausführungszeit nicht auswirken, da die Mengen der Endanwender vorberechnet sind. Jedoch sind solche Änderungen in der Praxis selten. Die dynamische Variante II hat wiederum den Vorteil, dass schon zur Modellierzeit weitere Prüfungen auf den

Bezeichnung	Beschreibung
Statisch	Bei Instanziierung eines Prozesses werden die Endanwender für alle Aktivitäten fix festgelegt.
Dynamisch I	Sobald eine Aktivität zur Ausführung ansteht, wird die Menge der Endanwender berechnet, die sich für diese Aktivität qualifizieren. Im Anschluss werden diese benachrichtigt, sofern sie am <i>PrMS</i> angemeldet sind.
Dynamisch II	Die potentiellen Endanwender einer Aktivität werden bereits zur Modellierzeit berechnet. Zur Ausführungszeit werden die vorberechneten Mengen potentieller Endanwender bei Aktivierung einer Aktivität verwendet. Es werden dazu aus der vorberechneten Menge diejenigen Endanwender benachrichtigt, die angemeldet sind.

Tabelle 5.3: Auswertungsvarianten für Endanwenderausdrücke

vorberechneten Mengen angestellt werden können. Diese Arbeit wird zeigen, dass solche Prüfungen die robuste Ausführung mobiler Aktivitäten steigern kann.⁶

Wie nun die Menge potentieller Endanwender der dynamischen Variante II ermittelt wird, zeigt Algorithmus 5.1. Er berechnet für alle Aktivitäten eines Prozessmodells *PM* die Menge potentieller Endanwender *PEA*(*n*).

Die mittels Algorithmus 5.1 berechneten Endanwendermengen für die Aktivitäten des Prozesses werden zur Ausführungszeit verwendet, um interaktive Aktivitäten zu Endanwendern zuzuteilen. Dazu wird bei Aktivierung einer Aktivität *n* vom *PrMS* überprüft, welche Endanwender aus *PEA*(*n*) am System angemeldet sind. Diesen wird die Aktivität *n* dann vom *PrMS* in ihre Arbeitslisten übertragen.

Ein weiterer Aspekt sind flexible Änderungen von Prozessen zur Laufzeit. Dazu sind sehr umfangreiche Prüfungen notwendig und es müssen sehr fortschrittliche Konzepte realisiert werden. Die in [Rei00, RD98] vorgestellten Ad-hoc Änderungen werden in dieser Arbeit übernommen. Ad-hoc Änderungen sind mit den vorgestellten Konzepten verträglich. Ferner sind die Konzepte dieser Arbeit auch mit den Konzepten der Schemaevolution, die für das ADEPT-Modell existieren [RHD98, Rin04, RRD04] verträglich. Zu beiden Aspekten wird dies im Verlauf der Arbeit nicht mehr explizit diskutiert, dennoch ist die Realisierung sowohl von Ad-hoc Änderungen als auch Schemaevolution in Kombination mit mobilen interaktiven Aktivitäten möglich.

Abschließend wird ein Ausführungsaspekt vorgestellt: die Ausführungshistorie von Prozessinstanzen. Diese protokolliert zu einer Aktivität die relevanten Ausführungsdaten. Generell werden Ausführungshistorien aus zwei Gründen angelegt. Erstens können damit bereits ausgeführte Aktivitäten wieder rückgängig gemacht werden. Zweitens lassen sich Ausführungshistorien mittels Process Mining [Aal11] analysieren, um Prozessabläufe zu optimieren.

Eine vereinfachte Ausführungshistorie zur Prozessinstanz aus Abbildung 5.12 könnte wie folgt aussehen:

Mögliche Ausführungshistorie der Prozessinstanz aus Abbildung 5.12
START(*a*, 1, *Marc Schickler*), END(*a*, 1), START(*b*, 1, *Johannes Schobel*), END(*b*, 1),

⁶siehe [CK08c, CK08a] für weitere Details bezüglich der Vor- und Nachteile der dynamischen Varianten

Algorithmus 5.1 : Bestimmung potentieller Endanwender der Aktivitäten eines Prozesses**Data** : Prozessmodell $PM = (N, E, \dots)$ *Endanwender*: Alle Endanwender des Organisationsmodells $EA1 - EA5$: $\forall n \in N$: Endanwenderausdrücke der Aktivitäten des Prozesses**Result** : $PEA(n)$: $\forall n \in N$: Mengen potentieller Endanwender der Aktivitäten des Prozesses

```

1 begin
2   /* Prozessaktivitäten  $n \in N$  werden in partieller Ordnung durchlaufen */
3   foreach  $n \in N$  do
4      $PEA(n) \leftarrow \emptyset$ ; /* Initialisierung */
5     /* Fallprüfung für Prozess-Constraints */
6     switch  $EA3(n) \vee EA4(n) \vee EA5(n)$  do
7       case  $EA3(n)$  referenziert Aktivitaet m
8         foreach Endanwender ea : autorisiert durch PEA(m) do
9           foreach Endanwender ea2 : autorisiert durch EA1(n)  $\vee$  EA2(n) do
10            if  $EA3(n) \wedge ea = ea2$  then
11               $PEA(n) \leftarrow PEA(n) \cup ea2$ 
12            end
13          end
14        end
15      case  $EA4(n) \vee EA5(n)$ 
16        foreach Endanwender ea : autorisiert durch EA1(n)  $\vee$  EA2(n) do
17           $PEA(n) \leftarrow PEA(n) \cup ea$ 
18        end
19      otherwise
20        foreach Endanwender ea : autorisiert durch EA1(n)  $\vee$  EA2(n) do
21           $PEA(n) \leftarrow PEA(n) \cup ea$ 
22        end
23      end
24    endsw
25  end
26 end

```

$START(b, 2, Johannes\ Schobel), END(b, 2), START(c, 1, Simon\ Mayer), END(c, 1),$
 $START(f, 1, Bernhard\ Miller), END(f, 1), START(g, 1, Stefan\ Briel),$
 $START(d, 1, Simon\ Mayer), END(d, 1), \dots$

Bei jedem Start einer Aktivität wird ein Eintrag *START* erzeugt, der den Bezeichner der Aktivität, den Iterationszähler der Aktivität und den ausführenden Endanwender enthält. Korrespondierend dazu wird nach Beendigung der Aktivität ein Eintrag *END* erzeugt. Letzterer erfasst mit Ausnahme des ausführenden Endanwenders und eines Zeitstempels (hier nicht dargestellt) die gleichen Daten wie der *START*-Eintrag. Einträge zu automatischen Aktivitäten werden nicht weiters betrachtet und können in [Rei00] nachgelesen werden.

5.5 Zusammenfassung

Dieses Kapitel hat grundlegende Aspekte des Prozess-Metamodells und der Endanwenderzu- teilung vorgestellt, die für diese Arbeit relevant sind. Dazu gehören einerseits die verwendeten Endanwenderausdrücke für interaktive Aktivitäten. Andererseits ist die gewählte Variante zur

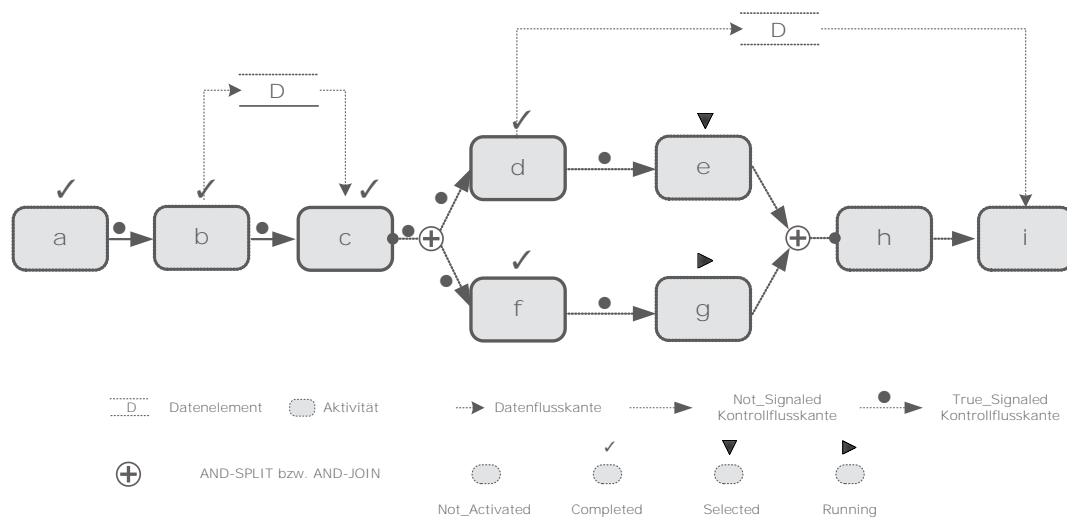


Abbildung 5.12: Prozessinstanz mit Ausführungszuständen

Zuteilung interaktiver Aktivitäten während der Ausführungszeit ein wichtiger Aspekt. Dazu wurde ein Algorithmus vorgestellt, der auch bei der robusten Ausführung mobiler Aktivitäten zum Einsatz kommt. Ferner wurden die Vor- und Nachteile der für diese Arbeit gewählten Zuteilungsvariante diskutiert.

Wer nicht gerne denkt, sollte wenigstens von Zeit zu Zeit seine Vorurteile neu gruppieren.

Luther Burbank (1849-1926)

6

Prozess-Architektur

Für existierende Ansätze für mobile Prozesse und mobile Architekturen lässt sich feststellen, dass eine vollständige Integration der entwickelten Konzepte in eine bestehende Architektur, welche die Grundlage jedes modernen Prozess-Management-Systems ist, nur selten verfolgt wird. Auch die Entwicklung einer neuen Architektur wird nur von wenigen Autoren betrachtet (siehe [Kre14] für eine Ausnahme). Da die Eigenschaften eines Smart-Mobilgeräts im Vergleich zu Desktop-Systemen eingeschränkt sind, ist sorgfältig zu bedenken, wie solche Mobilgeräte geeignet in eine Architektur integriert werden können.

Dieses Kapitel stellt die für die vorliegende Arbeit zugrunde gelegte Architektur vor und diskutiert wichtige Aspekte einer robusten Ausführung mobiler Aktivitäten. Die Prozess-Architektur ist ein wichtiger Bestandteil der in dieser Arbeit zugrunde gelegten Prozessumgebung. Das Kapitel gliedert sich wie folgt: Abschnitt 13.1 leitet die Prozess-Architektur her. Abschnitt 6.2 gibt dann einen Überblick zu dieser Prozess-Architektur und ihren Komponenten. Die Abschnitte 6.3 und 6.4 stellen den Prozess-Klienten und die Worklist vor. In Abschnitt 6.5 wird das Ausführungsprotokoll zwischen *PrMS* und Prozess-Klient eingeführt. Abschnitt 6.6 fasst das Kapitel zusammen.

6.1 Einleitung

Die Architektur eines *PrMS* bestimmt die angebotenen Möglichkeiten für eine Prozessunterstützung. Dabei ist anzustreben, dass die zur Verfügung gestellten Werkzeuge für die Prozess-Modellierung möglichst gut auf die Ausführung abgestimmt werden. Des Weiteren ist für eine Architektur zu überlegen, welche nicht-funktionalen Eigenschaften realisiert werden sollen. So kann Skalierbarkeit als wichtige Anforderung identifiziert werden, was entsprechende Konzepte bedingt. Bestenfalls ist die Architektur modular aufgebaut, sodass Komponenten jederzeit ausgetauscht werden können, etwa um neue Funktionen anzubieten, bestehende zu verändern oder nicht-funktionale Eigenschaften geeignet zu adressieren.

Referenzarchitekturen für *PrMS* [WMC94] helfen, die notwendigen Komponenten einer Architektur zu identifizieren. Die nachfolgend vorgestellte Architektur berücksichtigt sowohl Referenzarchitekturen als auch den aktuellen Stand der Forschung (z.B. [GG99, Jab97, WMC94,

RDRM⁺09a, Buc12, Wes07, AVH04, Kre14]). Sie dient ferner dazu, die in der Folge vorgestellten Konzepte praktisch zu realisieren, um deren Machbarkeit zu demonstrieren. Darüber hinaus werden Service-orientierte Architekturen [Erl05, BBP09] betrachtet, da hier die Integration von Web Services [ACKM04, BBR11] mit WS-BPEL [WCL⁺05, Erl04, Erl05] erfolgt, wobei Prozesse (auch Service-Orchestrierung genannt) eine große Rolle spielen. Zuletzt wurden existierende *PrMS* analysiert [NFN⁺09, SAP14b, IBM14a, KLL09].

6.2 Architekturkomponenten

Abbildung 6.1 illustriert die in dieser Arbeit entwickelte Prozess-Architektur. Tabelle 6.1 stellt die Komponenten der Architektur im Detail vor.

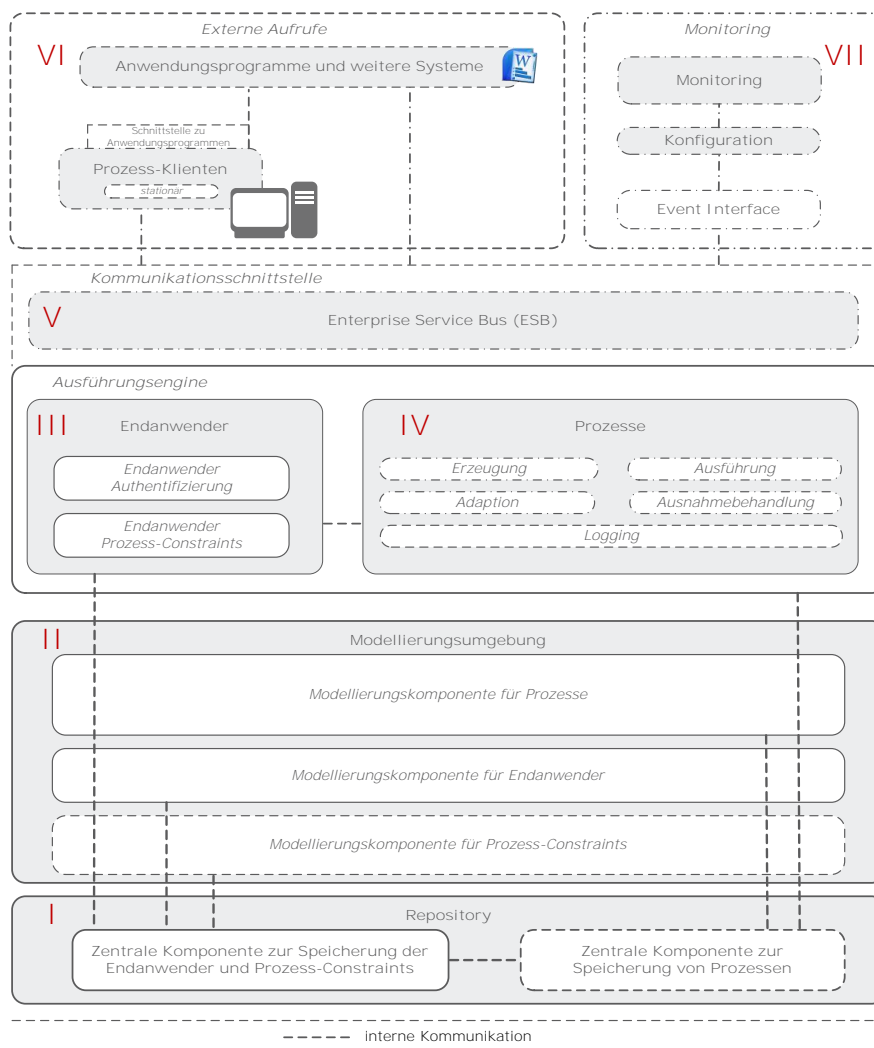


Abbildung 6.1: Architekturkomponenten im Überblick

	Architekturkomponenten	Beschreibung
<i>I</i>	Repository	Im zentralen Repository [BBP09, BBTR14] werden die modellierten Prozesse, Endanwendersausdrücke und Endanwender verwaltet und gespeichert (vgl. Kapitel 5).
<i>II</i>	Modellierungsumgebung	In der Modellierungsumgebung werden Prozesse definiert. Dies inkludiert die Modellierung von Endanwendern und Endanwendersausdrücken (vgl. Kapitel 5).
<i>III</i>	Ausführungseingine Endanwender	Diese Komponente prüft zur Ausführungszeit die Endanwendersausdrücke für Endanwender.
<i>IV</i>	Ausführungseingine Prozesse	Diese Komponente ist die zentrale Ausführungskomponente für Prozesse. Letztgenannte werden instanziiert, ausgeführt und bei Bedarf adaptiert [RW12]. Ebenso wird auf auftretende Ausnahmen (z.B. Crash eines Smart-Mobilgeräts) angemessen reagiert [PMR14]. Schließlich erzeugt die Komponente die Ausführungshistorie (sog. Log).
<i>V</i>	Kommunikationsschnittstelle	Es muss eine Kommunikationsschnittstelle für die Programme existieren, die vom <i>PrMS</i> aufgerufen werden sollen. Bezogen auf die zwei Typen einer Aktivität (automatische vs. interaktive) werden entweder die stationären Systeme von Endanwendern gerufen (interaktive Aktivitäten) oder weitere Systeme kontaktiert (automatische Aktivitäten). Diese Kommunikationsschnittstelle wird in dieser Arbeit über einen Enterprise Service Bus [BBP09] realisiert.
<i>VI</i>	Externe Aufrufe	Bei externen Aufrufen handelt es sich um die unter V vorgestellten Aufrufe. Die aufgerufenen Desktop-Anwendungen der Endanwender werden als Prozess-Klienten bezeichnet. Prozess-Klienten sind dem entsprechend der Klient-seitige Teil des <i>PrMS</i> , um interaktive Aktivitäten zu realisieren.
<i>VII</i>	Monitoring	Mittels Monitoring realisieren <i>PrMS</i> die Überwachung von Prozessen. Dazu werden die zu überwachenden Parameter festgelegt [BBP09, BBR13]. So kann die Anzahl an Ausnahmen protokolliert werden, um beim Überschreiten einer Ausnahmegrenze geeignete Gegenmaßnahmen einzuleiten. In diesem Kontext kommt das <i>Business Activity Monitoring</i> für fortschrittliche Analysen zum Einsatz [KH08, Sch11b]. Wir beschränken das Monitoring in der vorliegenden Arbeit auf die Überwachung der Parameter mobiler Aktivitäten.

Tabelle 6.1: Beschreibung der Architekturkomponenten

6.3 Prozess-Klient

Prozess-Klienten führen interaktive Aktivitäten aus. Sie sind der Klient-seitige Teil des *PrMS*, um interaktive Aktivitäten zu realisieren. Es handelt sich um aufwändige Anwendungsprogramme, die auf einem Desktop-System installiert werden und die gesamte Kommunikation mit dem *PrMS* durchführen. Da der Fokus auf Konzepten für interaktive mobile Aktivitäten liegt, ist eine differenzierte Sicht auf den Prozess-Klienten besonders wichtig (vgl. Abb. 6.2).

Prozess-Klienten bestehen im Kern aus drei Sub-Komponenten: (1) Der Authentifizierungsklient (vgl. Abb. 6.2⑤) ist dafür verantwortlich, dass sich ein Prozess-Klient am *PrMS* anmelden kann; (2) Der Worklist-Klient (vgl. Abb. 6.2⑥) verwaltet die von einem Endanwender

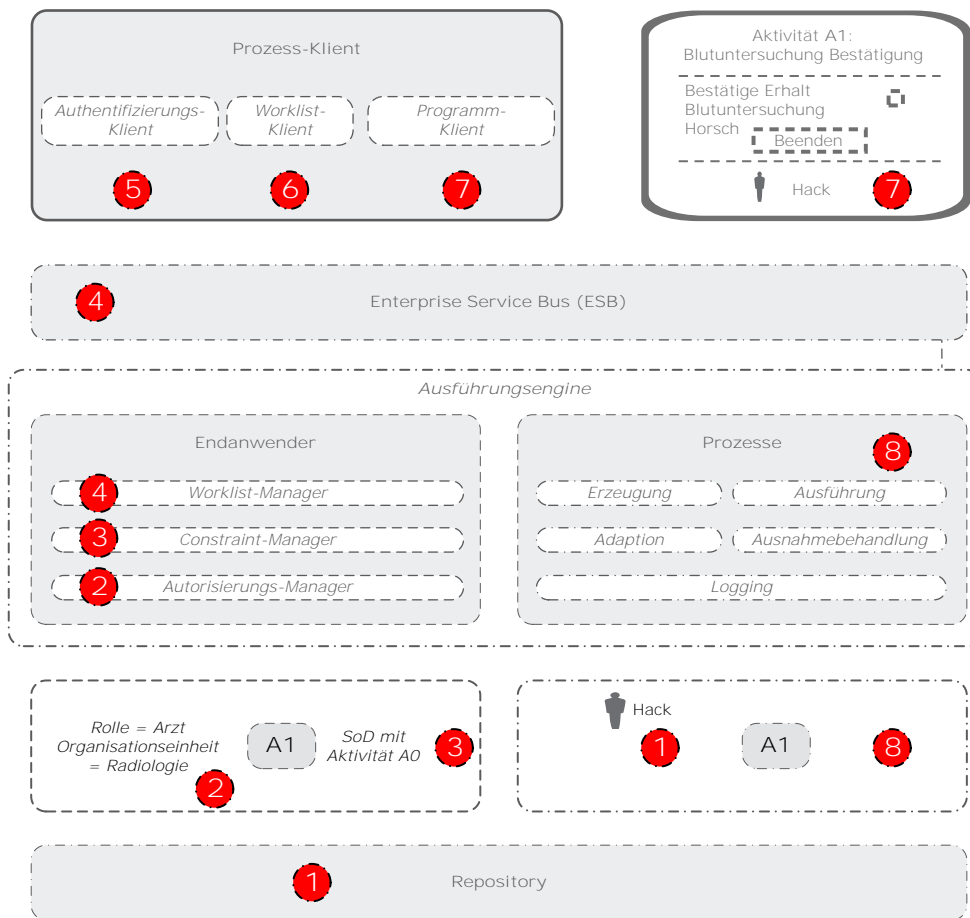


Abbildung 6.2: Prozess-Klient

angenommen Aktivitäten; (3) Der Programm-Klient (vgl. Abb. 6.2⑦) wiederum führt die Aktivitäten auf dem System des Endanwenders aus. Dieser regelt auch das Aufrufen weiterer Anwendungsprogramme, sofern notwendig. Die Funktionsweise des Prozess-Klienten bei Zuteilung einer Aktivität (vgl. Abb. 6.2①-⑧) zeigt Tabelle 6.2.

6.4 Worklists

In diesem Abschnitt werden die wichtigsten Aspekte von Worklists (dt. Arbeitslisten) vorgestellt. Worklists verwalten die Aktivitäten eines Endanwenders in seinem Prozess-Klienten. Generell sind dazu drei Aspekte wichtig. Erstens müssen die grundlegenden Einträge festgelegt werden, die eine Worklist enthält. Zweitens existieren Operationen, mit denen die Einträge der Worklist manipuliert werden können. Drittens existieren unterschiedliche technische Verfahren, um eine Worklist seitens des *PrMS* zu befüllen.

	Bezeichnung	Beschreibung
1	Aktivität wurde aktiviert	Aktivität A1 (vgl. Abb. 6.2⑦) steht zur Ausführung an, d.h. sie ist im Zustand <i>Activated</i> . Aus dem Repository werden die relevanten Daten abgerufen, um zu entscheiden, welche Endanwender sich für die Ausführung qualifizieren.
2&3	Autorisierungs- und Constraint Regeln	Aus dem Repository wird ermittelt, dass für A1 die Rolle <i>Arzt</i> und die Organisationseinheit <i>Radiologie</i> festgelegt wurde (vgl. Abb. 6.2②). Weiters existiert ein <i>Separation of Duties</i> Constraint zu Aktivität A0 (vgl. Abb. 6.2③). Mithilfe dieser Informationen bestimmen der Autorisierungs- und Constraint-Manager diejenigen Endanwender, die sich für die Ausführung von A1 qualifizieren.
4	Worklist-Manager	Der Worklist-Manager befüllt über die Kommunikationsschnittstelle die Worklists der sich für A1 qualifizierenden Endanwender.
5	Authentifizierungs-Klient	Der Authentifizierungs-Klient regelt die Anmeldung des Endanwenders beim <i>PrMS</i> . Sobald sich ein Endanwender erfolgreich am <i>PrMS</i> anmeldet, wird er bei der Aktivitätenenzuteilung berücksichtigt.
6	Worklist-Klient	Der Worklist-Klient nimmt Aktivitätenanfragen entgegen; im vorliegenden Fall ist dies die Anfrage für Aktivität A1. Übernimmt ein Anwender diese Aktivität, trägt sie der Worklist-Klient in die Worklist des Endanwenders ein. Aus den Listen aller übrigen Endanwender wird diese Aktivität durch den Worklist-Manager des <i>PrMS</i> wieder entfernt.
7	Programm-Klient	Der Programm-Klient übernimmt die Ausführung der Aktivität, d.h. die Ausführung von A1 im Beispiel. Dazu entscheidet er, ob weitere Programme aufgerufen werden müssen. Ferner tauscht er die notwendigen Prozess-Daten mit dem <i>PrMS</i> aus.
8	Aktivitätenausführung	Die Ausführungseingine führt A1 aus. Einerseits werden Daten zwischen Programm-Klient und Ausführungseingine übertragen, andererseits können Ausnahmebehandlungen oder Prozessadaptionen notwendig werden. Dazu wird ein aufwändiges Protokoll realisiert.

Tabelle 6.2: Prozess-Klient und Aktivitätenausführung

6.4.1 Worklist-Einträge

Worklists enthalten zwei Typen von Einträgen: Zur Ausführung anstehende Aktivitäten, für die sich noch kein Endanwender entschieden hat, sowie Aktivitäten, die ein Endanwender zur Ausführung übernommen hat. Erstere sind bei allen Endanwendern in der Worklist, die sich für die Ausführung der Aktivität qualifizieren. Diese werden als anstehende Aktivitäten (kurz: *QuActivities*; Qu=Queuing) bezeichnet. Übernommene Aktivitäten sind nur bei demjenigen Endanwender in der Worklist, der die Aktivität auch tatsächlich übernommen bzw. angenommen hat. Diese werden als übernommene (bzw. *meine*) Aktivitäten bezeichnet (kurz: *MyActivities*). Das *PrMS* entfernt bei Übernahme einer Aktivität durch einen Endanwender – mithilfe eines transaktionsorientierten Protokolls – die Aktivität aus allen anderen Worklists der für die Ausführung der Aktivität ebenfalls qualifizierten Endanwender. Das konkrete Protokoll dazu spielt im Kontext dieser Arbeit keine wesentliche Rolle und wird daher nicht weiter vorgestellt.

Die konkreten Worklist-Einträge (*Workitems*) entsprechen Tupeln mit folgenden sechs Kompo-

nenten: Es existiert ein (1) eindeutiger Bezeichner für Aktivitäten (*ID*) und (2) ein Statusfeld (*Status*), beides sind fakultative Angaben. Ferner gibt es die optionalen Komponenten zur (3) Beschreibung von Aktivitäten und zur Angabe von (4) Bearbeitungsdauern (*Duration*), d.h., Angaben dazu bis wann die Aktivität beendet sein muss. Hinzu kommt eine optionale Komponente zur Festlegung der (5) Priorität einer Aktivität. Basierend auf den jeweiligen Prioritäten lassen sich Aktivitäten einer Worklist gewichten. Schließlich wird gespeichert, in welcher (6) Iteration der Ausführung sich die Aktivität befindet (*Iteration*). Abbildung 6.3 zeigt abschließend Worklists für *MyActivities* und *QuActivities*.

Max Mustermann

Worklist

Meine
Aktivitäten

Anstehende
Aktivitäten

Beschreibung

Status

Zeitlimit

Röntgenbilder Frau Maier anschauen	gestartet	4h 30min
Bestätige Erhalt Blutuntersuchung Horsch	gestartet	13:30 Uhr
Befundung Frau Zemas	gestartet	-----
Laborwerte Herr Zweller anfordern	gestartet	0h 0min
-----	-----	-----

Max Mustermann

Worklist

Meine
Aktivitäten

Anstehende
Aktivitäten

Beschreibung

Zeitlimit

Gipsen Herr Toppas	4h 30min
OP-Aufklärung Frau Astop	16:15 Uhr
Abrechnung Herr Muster	-----
Entlassung Frau Gaiser	1h 40min
-----	-----

(a) Max Mustermann:
 Meine Aktivitäten
 (MyActivities)

(b) Max Mustermann:
 Anstehende Aktivitäten
 (QuActivities)

Abbildung 6.3: Worklist-Beispiele (vereinfacht)

Angemerkt sei, dass Angaben zur Priorität und maximaler Zeitdauer für die Ausführung mobiler Aktivitäten für deren robuste Ausführung eine wichtige Rolle spielen.

6.4.2 Worklist-Operationen

Für eine Worklist existieren Operationen, mit denen die Komponenten eines Eintrags verändert werden können. Die eingesetzten Operationen sind Tabelle 6.3 zu entnehmen. Ferner zeigt Tabelle 6.3, ob das *PrMS*, der Programm-Klient oder beide eine Operation anwenden dürfen.

6.4.3 Worklist-Befüllung

Für die Übertragung von Aktivitäteneinträgen vom *PrMS* zum Worklist-Klient und umgekehrt, gibt es zwei unterschiedliche Verfahren. Entweder überträgt das *PrMS* aktiv Einträge oder

Operation	Anwendung auf	Beschreibung	PrMS	Programm-Klient
Add	<i>MyActivity</i>	Hinzufügen einer Aktivität zu meinen Aktivitäten	✓	X
Del	<i>MyActivity</i>	Löschen einer meiner Aktivitäten	✓	X
Get	<i>MyActivities</i>	Abfragen meiner Aktivitäten	✓	✓
Add	<i>QuActivity</i>	Hinzufügen einer anstehenden Aktivität	✓	X
Del	<i>QuActivity</i>	Löschen einer anstehenden Aktivität	✓	X
Get	<i>QuActivities</i>	Abfragen aller anstehenden Aktivitäten	✓	✓
GetPrio	<i>MyActivity</i>	Abfragen der Priorität einer meiner Aktivitäten	✓	✓
SetPrio	<i>MyActivity</i>	Setzen der Priorität einer meiner Aktivitäten	✓	✓
GetPrios	<i>MyActivities</i>	Abfragen der Prioritäten meiner Aktivitäten	✓	✓
SortPrios	<i>MyActivities</i>	Sortieren meiner Aktivitäten gemäß Priorität (auf- oder absteigend)	✓	✓
GetPrio	<i>QuActivity</i>	Abfragen der Priorität einer anstehenden Aktivität	✓	✓
SetPrio	<i>QuActivity</i>	Setzen der Priorität einer anstehenden Aktivität	✓	✓
GetPrios	<i>QuActivities</i>	Abfragen der Priorität aller anstehenden Aktivitäten	✓	✓
SortPrios	<i>QuActivities</i>	Sortieren aller anstehenden Aktivitäten gemäß Priorität (auf- oder absteigend)	✓	✓

Tabelle 6.3: Worklist-Operationen

Eintrag-Updates an den Worklist-Klienten (sog. Push-Verfahren)¹. Oder der Worklist-Klient fragt regelmäßig beim *PrMS* nach, ob neue oder veränderte Einträge existieren (sog. Push-Verfahren). Beide Verfahren haben ihre Vor- und Nachteile. Diese können in [Rei00] im Detail nachgelesen werden. Für die robuste Ausführung mobiler Aktivitäten wird das Push-Verfahren gewählt. Dies ist durch folgenden Umstand begründet: Die Prozess-Klienten laufen bei der Ausführung mobiler Aktivitäten auf Smart-Mobilgeräten ab, die wiederum anfälliger als Desktop-Systeme sind. Um möglichst frühzeitig Probleme angemeldeter Smart-Mobilgeräte (z.B. eine Offline-Situation) zu erkennen, ist die Variante des Push-Verfahrens besser für die Ausführung mobiler Aktivitäten geeignet. Mit dem Push-Verfahren werden alle Smart-Mobilgeräte konzentriert und zum gleichen Zeitpunkt angefragt. Beim Push-Verfahren dagegen lässt sich die Anfrage wesentlich schlechter koordinieren. Aus diesem Grund wird in der Folge das Push-Verfahren zum Befüllen von Worklists zugrunde gelegt.

6.5 Protokoll der Aktivitätsausführung

Bei der Ausführung interaktiver Aktivitäten muss ein *PrMS* mit dem Prozess-Klienten in sicherer Art und Weise eine Reihe von Daten austauschen. Dazu werden Protokolle (sog. Ausführungsprotokolle) eingesetzt. Deren Leistungsfähigkeit ist einerseits davon abhängig, welche Ausführungszustände berücksichtigt werden. Sollen zum Beispiel nur Daten beim Start und Ende einer Aktivität ausgetauscht werden, ist die resultierende Protokollvariante weniger leistungsfähig als diejenige, die auch einen Datenaustausch während der Ausführung der Aktivität erlaubt. Andererseits ist die Leistungsfähigkeit davon abhängig, welche konkreten Daten übertragen werden sollen.

Nach unserem Kenntnisstand werden Protokolle für mobile interaktive Aktivitäten bisher nicht ausreichend betrachtet. D.h. es existieren wenige verwandte Ansätze, die sich generell mit Ausführungsprotokollen zwischen Prozess-Klient und *PrMS* auseinandersetzen [AGA⁺95]. Da ein

¹Das Push-Verfahren wird als Broadcasting-Variante realisiert [Cov72].

solches Protokoll einen grundlegenden Beitrag zur robusten Ausführung mobiler Aktivitäten darstellt, wird es nachfolgend explizit betrachtet.

Abbildung 6.4 zeigt das zugrunde gelegte Ausführungsprotokoll im Detail. Hieran sind seitens des *PrMS* das Repository und die Ausführungseengine beteiligt (vgl. Abb. 6.4①). Seitens des Prozess-Klienten sind der Worklist-Klient, der Programm-Klient und das Anwendungsprogramm, welches zur Ausführung der Aktivität aufgerufen wird, beteiligt (vgl. Abb. 6.4②).

Die einzelnen Phasen des Ausführungsprotokolls stellt Tabelle 6.4 im Detail vor. Hier wird vereinfachend nur vom *PrMS* und Prozess-Klienten gesprochen, ohne die Teilkomponenten weiter zu unterscheiden (vgl. Abb. 6.4). Ferner wird das Ausführungsprotokoll nur für den Fall betrachtet, dass die Aktivität ordnungsgemäß ausgeführt werden kann. Die weiteren Fälle müssen nicht explizit behandelt werden, da sie den generellen Protokollablauf nicht verändern. Lediglich ein Fall kann weitere Schritte erfordern: Reagiert das vom Prozess-Klienten aufgerufene Anwendungsprogramm nicht mehr, muss dies explizit durch das Ausführungsprotokoll behandelt werden. Der letztgenannte Fall wird auch in [AGA⁺95] adressiert; er kann auch für mobile interaktive Aktivitäten angewandt werden.

Phase	Beschreibung
1	Der Endanwender hat eine Aktivität (<i>MyActivity</i>) aus seiner Worklist ausgewählt und möchte diese ausführen. Dies wird dem <i>PrMS</i> durch den Prozess-Klienten unter Angabe der ID der Aktivität mitgeteilt.
2&3	Das <i>PrMS</i> ändert den Status der Aktivität auf <i>Started</i> (vgl. Abb. 5.5).
4	Das <i>PrMS</i> bestätigt dem Prozess-Klienten, dass die Aktivität ausgeführt werden darf. Ferner überträgt das <i>PrMS</i> notwendige Prozessdaten für die auszuführende Aktivität. Das sind zum Beispiel Daten, die bereits ausgeführte Aktivitäten erzeugt haben und die von der nun auszuführenden Aktivität verwendet werden sollen. Hier sei angemerkt, dass der Prozess-Klient in Phase 4 auch die Nachricht vom <i>PrMS</i> erhalten kann, dass die Aktivität nicht gestartet werden darf. So kann es passieren, dass die Aktivität zwischenzeitlich zurückgesetzt wurde, weil der ganze Prozess abgebrochen wurde.
5&6	Die Aktivität wird vom Prozess-Klienten gestartet. Eventuell notwendige Anwendungsprogramme werden zur Ausführung aufgerufen.
7 – 12	Sollten während der Aktivitätsausführung weitere Daten vom <i>PrMS</i> benötigt werden, wird dies durch die Phasen 7-12 zwischen <i>PrMS</i> und Prozess-Klient realisiert.
13&14	Der Programm-Klient meldet dem <i>PrMS</i> , dass die Ausführung der Aktivität beendet ist. Darüber hinaus schickt es die veränderten oder neu erzeugten Daten an das <i>PrMS</i> .
15&16	Das <i>PrMS</i> ändert den Status der Aktivität auf <i>Completed</i> (vgl. Abb. 5.5). Weiters archiviert das <i>PrMS</i> die erzeugten oder veränderten Daten (vgl. <i>Archived</i> Abb. 5.5).
17	Das <i>PrMS</i> meldet an den Prozess-Klienten, dass die Aktivität ordnungsgemäß beendet wurde.
18	Der Prozess-Klient bestätigt gegenüber dem <i>PrMS</i> , dass die Aktivität ordnungsgemäß beendet wurde.

Tabelle 6.4: Phasen des Ausführungsprotokolls

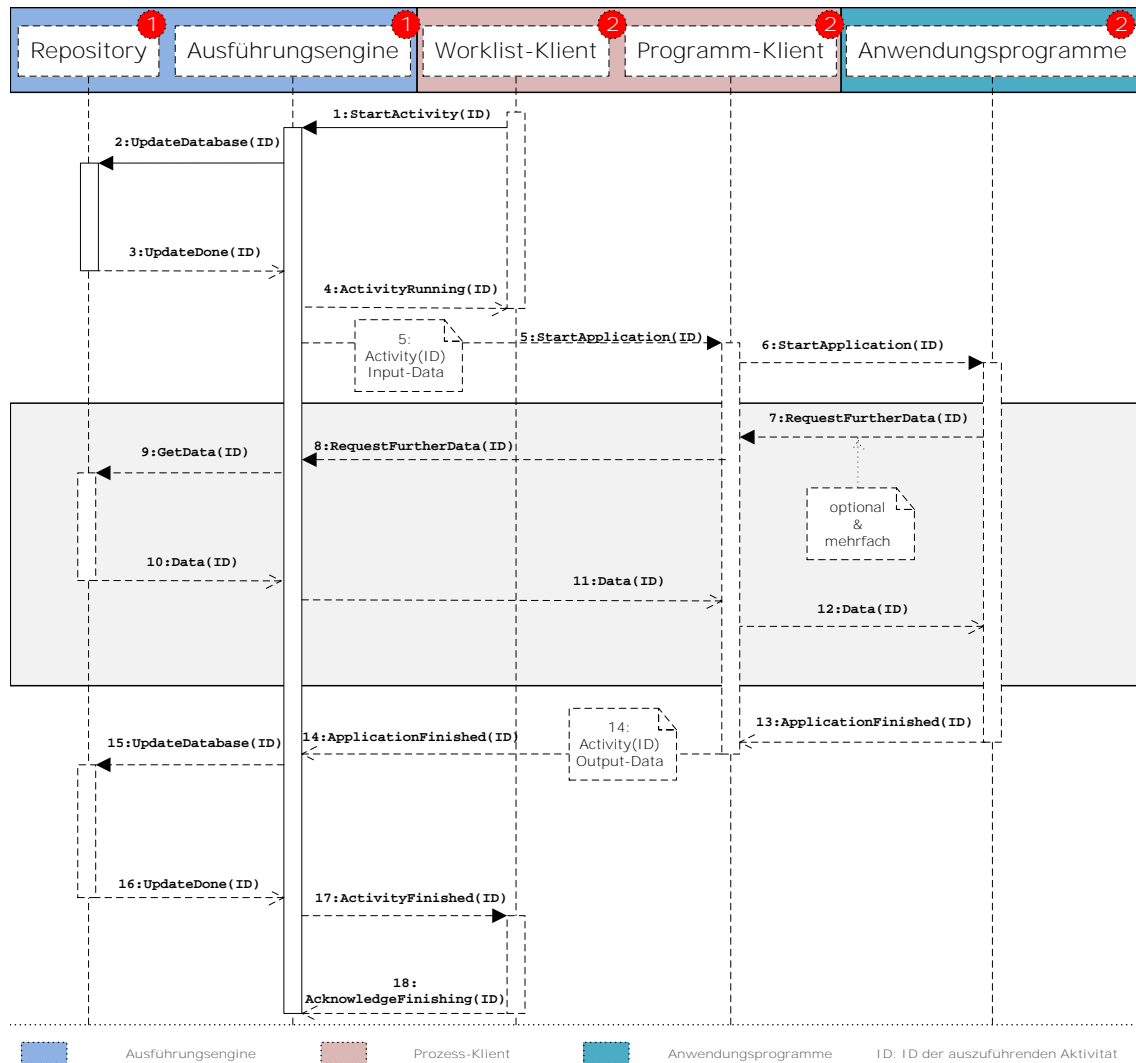


Abbildung 6.4: Protokoll der Aktivitätsausführung

6.6 Zusammenfassung

Dieses Kapitel hat die Grundzüge der Prozess-Architektur vorgestellt, die dieser Arbeit zugrunde gelegt wird. Insbesondere wurde der Prozess-Klient vorgestellt, da dieser im Kontext der robusten Ausführung mobiler Aktivitäten wichtig ist. Ergänzt wurde diese Darstellung durch die Vorstellung des angewandten Worklist-Konzepts sowie des zwischen Prozess-Klienten und *PrMS* verwendeten Protokolls zur Ausführung von Aktivitäten.

Teil III

Technische Lösung

Um klar zu sehen, genügt oft ein Wechsel der Blickrichtung.

Antoine de Saint-Exupéry (1900-1944)

7

Mobiler Kontext

Das vorliegende Kapitel behandelt den mobilen Kontext, der für die Modellierung, Erzeugung und Ausführung mobiler Aktivitäten benötigt wird. Ohne die Nutzung eines mobilen Kontexts wäre es nicht möglich, eine robuste Ausführung mobiler Aktivitäten zu gewährleisten. Beispiele für Kontextinformationen sind der Ort an dem eine mobile Aktivität ausgeführt werden soll oder der Ort an dem sich ein mobiler Endanwender aktuell befindet. Ist zum Beispiel der Ort der Ausführung einer mobilen Aktivität bekannt, kann man diejenigen Endanwender bevorzugt zuteilen, die sich nahe diesem Ausführungsort aufhalten. Besser geeignete mobile Endanwender können die Aktivität mit höherer Wahrscheinlichkeit rascher ausführen, was wiederum Akkuleistung spart und Ausnahmequellen minimiert.

In Kapitel 2 wurden mobile Aktivitäten bereits abstrakt definiert. Hier sei nochmals erwähnt, dass die vorliegende Arbeit den Fokus auf mobile interaktive Aktivitäten legt, d.h. Aktivitäten die von einem mobilen Endanwender ausgeführt werden. Im Folgenden sprechen wir daher *nur* noch von mobilen Aktivitäten.

Um den mobilen Kontext zu erfassen, wird in der Folge ein Parameterkatalog entwickelt, der sich in vier Bereiche untergliedert. Letzteres ermöglicht eine differenzierte Betrachtung des Kontexts. Sie wird erforderlich, um den verschiedenen Facetten mobiler Aktivitäten gerecht zu werden. Es wird gezeigt, dass ohne adäquate Berücksichtigung des Parameterkatalogs die Modellierung, Erzeugung und Ausführung mobiler Aktivitäten vor dem Hintergrund der geforderten Robustheit nicht gewährleistet werden kann. Mit der Parametrisierung mobiler Aktivitäten wird einerseits das Ziel verfolgt, die Zuteilung mobiler Endanwender zu mobilen Aktivitäten geeignet durchzuführen, etwa mithilfe von Informationen dazu welcher mobile Endanwender dem geplanten Ausführungsort der mobilen Aktivität am nächsten ist. Andererseits sollen Ausnahmesituationen proaktiv erkannt werden können (z.B. frühzeitiges Erkennen der Nichterreichbarkeit eines Smart-Mobilgeräts), um spezifische Ausnahmebehandlungsmaßnahmen einzuleiten.

Kapitel 7 gliedert sich wie folgt: Abschnitt 7.1 motiviert den mobilen Kontext, während Abschnitt 7.2 die zu seiner Ermittlung angewandte Methodik vorstellt. Abschnitt 7.3 führt den Parameterkatalog ein und diskutiert die einzelnen Parameter im Detail. Weitergehende Aspekte zum mobilen Kontext behandelt Abschnitt 7.4. In Abschnitt 7.5 wird analysiert, inwieweit der eingeführte Parameterkatalog als Basis für den mobilen Kontext als geeignet angesehen

werden kann. Weiters wird der zu erwartende Spezifikations- und Verwaltungsaufwand in Verbindung mit der Einführung eines solchen Katalogs diskutiert. Abschnitt 7.6 fasst das Kapitel zusammen.

7.1 Einleitung

Anhand des in Abbildung 7.1 dargestellte Prozessfragments soll motiviert werden, warum eine Parametrisierung mobiler Aktivitäten wichtig ist und ein umfassender Parameterkatalog zur Herstellung eines mobilen Kontexts notwendig wird. Abbildung 7.1 zeigt zum Beispiel, dass bei einem Verbindungsabbruch des Smart-Mobilgeräts der Prozessablauf (siehe die Prozesselemente in der gefärbten Fläche) nicht ordnungsgemäß fortgeführt werden kann, da infolge des Verbindungsabbruchs eine Deadlock-Situation [Dea14] entsteht. Für diesen Fall benötigt man zwingend eine Ausnahmebehandlung oder man bricht die mobile Aktivität ab; andernfalls bleibt der Prozess blockiert.

Bricht man die mobile Aktivität im Zuge der Ausnahmebehandlung ab, ergeben sich ggf. negative Effekte: Wie in Abbildung 7.1 zu sehen ist, schreibt die mobile Aktivität auch Daten, die von nachfolgenden Aktivitäten benötigt werden. Daher würde im gezeigten Beispiel bei Abbruch der mobilen Aktivität eine Folgeausnahme im weiteren Prozessablauf resultieren.

Generell lässt sich argumentieren, dass eine solche Ausnahme im Kontext einer mobilen Aktivität keine neuartige Situation darstellt, verglichen zu Ausnahmen nicht mobil ausgeführter Aktivitäten. Bezogen auf das Prozessfragment aus Abbildung 7.1 etwa wäre der gekennzeichnete Prozessbereich auch dann nicht weiter ausführbar, wenn die fehlgeschlagene Aktivität nicht mobil, sondern stationär ausgeführt werden würde. Bezogen auf den Prozessablauf (und auch den Datenfluss) trifft diese Aussage zu. Die Situation ändert sich jedoch grundlegend, wenn man die konkrete Ausnahmebehandlung und nicht nur den weiteren Verlauf des Prozesses betrachtet. Einen Verbindungsabbruch bei stationären Systemen zu berücksichtigen, ist im Allgemeinen unnötig. Ferner ist es unwahrscheinlich, dass ein stationäres Desktop-System überhaupt nicht mehr reagiert. Daher sind die Voraussetzungen sehr unterschiedlich, die man für die Ausnahmebehandlung einer mobil bzw. einer nicht mobil ausgeführten Aktivität zugrunde legen muss.

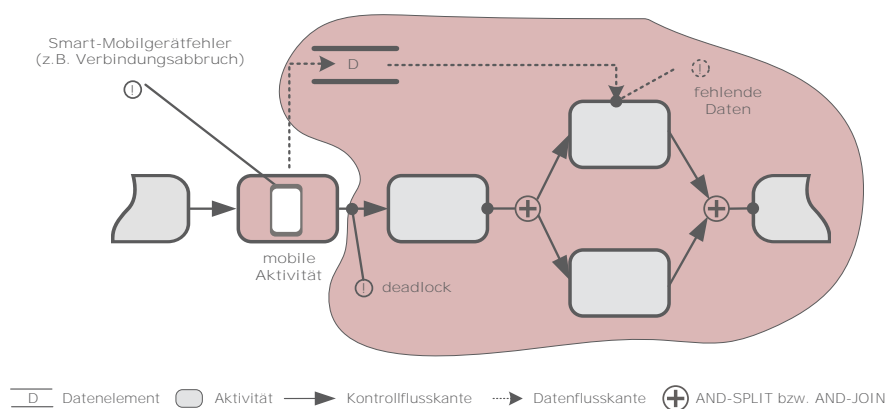


Abbildung 7.1: Prozessfragment mit mobiler Aktivität

Generell ist die Behebung von Ausnahmen bei der Ausführung mobiler Aktivitäten komplexer. Das folgende Beispiel soll dies verdeutlichen: Ist eine nicht mobil ausgeführte Aktivität, respektive das ausführende stationäre System, über einen längeren Zeitraum physisch nicht erreichbar, kann davon ausgegangen werden, dass dieses System einen Defekt hat und entweder ein anderer Endanwender die Aktivität ausführen oder der gleiche Endanwender ein anderes stationäres System verwenden muss.¹ Tritt die identische Ausnahme bei einer mobilen Aktivität auf, könnte das ausführende Smart-Mobilgerät auch nur vom Netz getrennt worden sein, weil sich der mobile Endanwender aus dem Netzbereich herausbewegt hat. Geht er anschließend wieder Online und wurde die Aktivität zwischenzeitlich an jemand anders übergeben, müssten eventuell schon geschriebene Daten des ersten mobilen Endanwenders verworfen werden.

Die Nichtverfügbarkeit eines mobilen Endanwenders aufgrund von Ausnahmen des Smart-Mobilgeräts wiegt aus einem weiteren Grund schwerer als bei nicht mobil ausgeführten Aktivitäten. Da mobile Endanwender meist individuell konfigurierte Smart-Mobilgeräte mit sich tragen oder sich nur an bestimmten Orten aufhalten, die für die Ausführung notwendig sind, stellt jeder für die Ausführung nicht mehr verfügbare mobile Endanwender eine potentielle Schwächung für die Ausführung weiterer mobiler Aktivitäten dar. Fall es keine geeigneten mobilen Endanwender mehr gibt, ist eine weitere Ausführung des Prozesses nicht mehr möglich. Auf Basis dieser Ausführungen lassen sich zwei grundsätzliche Ansätze identifizieren, um die Ausführung mobiler Aktivitäten robuster zu gestalten:

1. Ansatz 1 (*Proaktive Ausnahmebehandlung*): Für die Ausnahmebehandlung mobiler Aktivitäten wird eine sog. Vorphase (noch vor der Ausführung) etabliert, die entscheidend mit zur Robustheit beiträgt. Es sollen proaktiv nur diejenigen mobilen Endanwender zur Ausführung einer mobilen Aktivität ausgewählt werden, die mit einer geringeren Wahrscheinlichkeit als andere eine Ausnahme verursachen.
2. Ansatz 2 (*Reaktive Ausnahmebehandlung*): Die Ausnahmebehandlung mobiler Aktivitäten wird speziell für die mobile Umgebung konzipiert und kommt erst dann reaktiv zum Einsatz, wenn eine Ausnahme auftritt.

Wir verwenden beide Ansätze für die robuste Ausführung mobiler Aktivitäten. Dazu wird ein Rahmenwerk für die Eigenschaften mobiler Aktivitäten bzw. den mobilen Kontext entwickelt, das als Grundlage den bereits erwähnten Parameterkatalog verwendet. Parameter dieses Katalogs werden in zwei Phasen angewandt:

1. Die erste Phase betrifft die Ausführung mobiler Aktivitäten. Die Parameter werden eingesetzt, um die am besten geeigneten mobilen Endanwender für die Ausführung mobiler Aktivitäten zu wählen. Ziel ist es, die Ausfallquote für die letztlich zugeteilten mobilen Endanwender möglichst gering zu halten. Dies entspricht Ansatz 1, d.h. einer proaktiven Ausnahmebehandlung.
2. Die zweite Phase betrifft die robuste Behandlung eingetretener Ausnahmen bei der Ausführung mobiler Aktivitäten. Hier wird sichergestellt, dass ein Prozess in korrekter Weise und ohne Folgeausnahmen fortgeführt werden kann. Dies entspricht Ansatz 2, d.h. einer reaktiven Ausnahmebehandlung.

¹Hier wird angenommen, dass ein stationäres System über ein stabiles Netzwerk angebunden ist.

Die Umsetzung von Ansatz 1 wird im Folgenden als *Kontextbezogene mobile Endanwenderzuteilung (KMEZ)* bezeichnet, die von Ansatz 2 als *Ausnahmebehandlungsstrategie mobiler Aktivitäten (ABSM)*.

7.2 Forschungsmethodik

Dieser Abschnitt stellt die Methodik des Parameterkatalogs des mobilen Kontexts vor (vgl. Abb. 7.2). Da Letzterer grundlegend für die geforderte robuste Ausführung mobiler Aktivitäten ist, sollte er möglichst viele Anwendungsfälle abdecken und generisch sein. Die zur Bestimmung des Parameterkatalogs angewandte Methodik ist in Abbildung 7.2 zu sehen.

Es sollen geeignete Parameter zusammengefasst werden, die für eine kontextbezogene mobile Endanwenderzuteilung und Ausnahmebehandlungsstrategie genutzt werden können (vgl. Abb. 7.2). Um diese Parameter zu bestimmen, wurden zunächst Fallstudien in verschiedenen Anwendungsdomänen und danach eine umfangreiche Literaturrecherche durchgeführt (vgl. Abb. 7.2①).²

Generell existieren nur wenige Ansätze, die sich mit der Parametrisierung mobiler Aktivitäten befassen. [RS03] untersucht, inwiefern durch geeignete Parametrisierung mobiler Aktivitäten die ausführenden Endanwender möglichst in Sichtweite agieren können, um spezialisierte Protokolle für einen Austausch von Daten vor Ort (d.h. mit einem Peer-to-Peer-Protokoll) nutzen zu können. [Lac13] wiederum betrachtet einen Modellierungsansatz für mobile Parameter im Kontext von BPMN-basierten Prozessen. Ausführungsaspekte werden in dieser Arbeit nicht betrachtet.

Auf Grundlage der Literaturrecherche und Fallstudien waren zur Bestimmung des mobilen Kontexts sechs weitere Phasen durchzuführen (vgl. Abb. 7.2②):

1. In der ersten Phase wurden Parameterkategorien identifiziert. Es hat sich herausgestellt, dass sich jeder identifizierte Parameter stets einer von vier Kategorien (vgl. Abb. 7.2I-IV) zuordnen lässt. Durch diese Kategorisierung lässt sich der resultierende Katalog strukturierten, sodass er einfacher verwaltet werden kann. Weiters hilft die Kategorisierung bei der Entwicklung von Werkzeugen zur Modellierung von Parametern.
2. In der zweiten Phase wurden aus den Ergebnissen der Fallstudien und Literaturrecherche für jede Kategorie relevante Parameter und deren Abhängigkeiten identifiziert. Die Erfassung der Abhängigkeiten zwischen den Parametern, diente wiederum dem Ziel der robusten Ausführung mobiler Aktivitäten, da die Abhängigkeiten in der Folge so durch Algorithmen bewertet werden können. Insgesamt wurden **18** Parameter identifiziert, deren Aufteilung in die vier Kategorien in Abbildung 7.2 gezeigt wird. Abschließend sei erwähnt, dass bei der Identifizierung der **18** Parameter der Aspekt der Akzeptanz explizit berücksichtigt wurde, um die Eignung eines Parameters abschließend bewerten zu können. Diese Akzeptanz zeigt an, ob die auf Basis des Parameters abgeleiteten Maßnahmen hilfreich für einen Endanwender sind. Dies stellt eine explizite Anforderung (siehe Kapitel 2) dieser Arbeit dar.
3. In den dritten Phase war für jeden Parameter zu bestimmen, wann sein Wert festgelegt werden soll. Grundsätzlich kommen zwei Zeitpunkte infrage: (1) Entweder wird ein Para-

²Beide Maßnahmen können im Detail in Kapitel 2 nachgelesen werden.

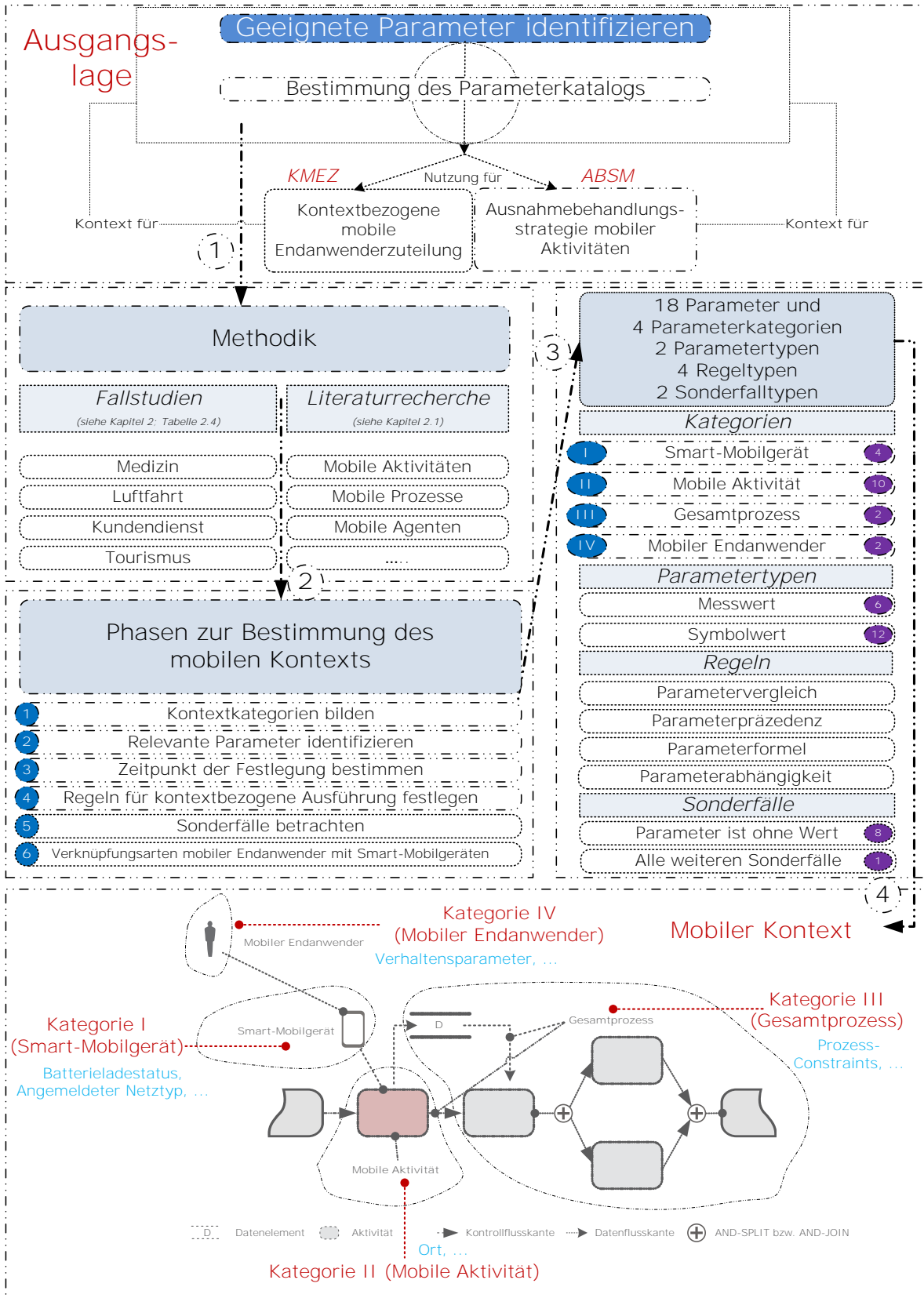


Abbildung 7.2: Angewandte Methodik für mobilen Kontext

meter zur Modellierzeit eines Prozesses für dessen mobile Aktivitäten festgelegt oder aber (2) zur Ausführungszeit. Weiters muss unterschieden werden, ob der Parameterwert von einem Endanwender manuell festgelegt werden muss oder vom System berechnet bzw. gemessen wird. In den beiden letztgenannten Fällen kann die Festlegung wiederum entweder zur Modellier- oder Ausführungszeit erfolgen.

Generell sollte vermieden werden, dass ein Parameter zur Ausführungszeit manuell festgelegt werden muss. So müsste er für jede Prozessinstanz wiederholt spezifiziert werden. Da zur Ausführungszeit – aufgrund potentiell vieler Prozessinstanzen – viele Aspekte zu berücksichtigen sind, wäre eine manuelle Festlegung zudem zu komplex.

Zu welchem Zeitpunkt ein Parameterwert festgelegt werden muss, wird auf Basis seines Typs entschieden. Hierzu konnten aus den Fallstudien und der Literaturrecherche zwei Typen identifiziert werden: Entweder (1) wird ein Parameter symbolisch definiert (*Symbolwert*) oder er wird (2) gemessen (*Messwert*). *Symbolwerte* werden in der Literatur genannt, um Parameter über abstrakte Symbole zu definieren [BD05, HA95, Dec11]. So können zum Beispiel zum symbolischen Parameter *Ausführungsort* zu einer mobilen Aktivität die abstrakten Symbole *OP* und *Aufwachraum* definiert werden. Parameter vom Typ *Symbolwert* haben sich als effektiv für die Beschreibung des mobilen Kontexts erwiesen. *Messwerte* werden hingegen vom *PrMS* bestimmt. Die Bestimmung erfolgt entweder durch das *PrMS* selbst oder angebundene Systeme wie Sensoren, etwa um den Ladezustand des Akkus eines Smart-Mobilgeräts zu bestimmen. Parameter vom Typ *Symbolwert* werden zur Modellierzeit manuell festgelegt. Dagegen werden Parameter vom Typ *Messwert* automatisch ermittelt. Letzteres kann sowohl zur Modellier- als auch Ausführungszeit erfolgen. Da *Symbolwerte* lediglich zur Modellierzeit festgelegt werden, entfällt der Aufwand einer evtl. manuellen Festlegung zur Ausführungszeit. Schließlich können *Symbolwerte* zur Ausführungszeit nicht geändert werden.

Von den insgesamt **18** identifizierten Parametern (siehe Abb. 7.2③) sind **12** vom Typ *Symbolwert* und **6** vom Typ *Messwert*.

4. In der vierten Phase wurden Regeln auf Basis des Parameterkatalogs festgelegt, die von der kontextbezogenen mobilen Endanwenderzuteilung und der mobilen Ausnahmebehandlungsstrategie genutzt werden können. Dazu wurden **4** Regeltypen identifiziert. Entweder (1) werden zwei Parameter miteinander verglichen (Parametervergleich), etwa $Ort(\text{Smart-Mobilgerät}) = Ort(\text{Mobile Aktivität})$ oder (2) es werden Präzedenzen zwischen ihnen festgelegt (Parameterpräzedenzen), etwa dass der Ort eines mobilen Endanwenders bei der Zuteilung einer mobilen Aktivität stärker bindet als die Akkuleistung seines Smart-Mobilgeräts. Weiters (3) können Berechnungen auf Basis eines oder mehrerer Parameter definiert werden (Parameterformel). Dadurch lassen sich weitere Parameter herleiten, etwa durch Gewichtung von Einzelparametern. Dies wiederum ermöglicht es, Szenarien spezifisch zu berücksichtigen, etwa um die Akkuleistung zum angemeldeten Netztyp eines Smart-Mobilgeräts in Bezug zu setzen. Der letzte Regeltyp definiert (4) Abhängigkeiten zwischen Parametern (Parameterabhängigkeit). Beispielsweise ergibt sich aus dem Ausführungsort einer mobilen Aktivität der Einzugsbereich für die mobilen Endanwender, die für die Ausführung der mobilen Aktivität geeignet erscheinen.
5. In der fünften Phase wurden Sonderfälle für die hergeleiteten Parameter betrachtet. Grundsätzlich werden zwei Arten von Sonderfällen unterschieden. Zuerst geht es um die Frage, welche Auswirkungen resultieren, wenn ein Parameter keinen Wert hat. Im Fall des Symbolwertes bedeutet dies, dass er manuell nicht festgelegt wird; im Fall des Mess-

werts wiederum konnte er nicht ermittelt werden. Zur zweiten Art der Sonderfälle gehören weitere Sonderfälle, die nachfolgend vorgestellt werden.

6. In der sechsten Phase wurde untersucht, ob bei der Integration von Endanwendern in ein *PrMS* die Hinzunahme von Smart-Mobilgeräten zu weiteren Auswirkungen führt. Dazu konnten zwei grundlegende Sichtweisen identifiziert werden, die in Abbildung 7.3 illustriert werden. Entweder kann sich ein Endanwender nur auf einem Smart-Mobilgerät exklusiv anmelden (vgl. Abb. 7.3 *oben*) oder es ist ihm gestattet, das Smart-Mobilgerät zu wechseln. Letzteres bedeutet, dass er sich auf beliebigen Smart-Mobilgeräten anmelden kann, aber dennoch zeitgleich immer nur auf einem Gerät.

Dass sich die beiden Sichtweisen unterschiedlich auswirken können auf die Verfügbarkeit von Endanwendern zeigt Abbildung 7.3. Die linke Seite zeigt die beiden Sichtweisen, während rechts dargestellt wird, wie sich ein Gerätefehler auf diese Sichtweisen auswirkt: Die beiden mobilen Endanwender "Schickler" und "Schobel" sind jeweils an einem Smart-Mobilgerät (Geräte 1 bzw. 3) angemeldet. Nun tritt bei beiden Geräten eine Ausnahme auf. Für den mobilen Endanwender "Schickler" bedeutet dies, dass er ab sofort für das *PrMS* solange nicht mehr zur Verfügung steht, bis er sein Smart-Mobilgerät (1) wieder aktiviert. Im Gegensatz dazu meldet sich der mobile Endanwender "Schobel" an einem anderen Smart-Mobilgerät (4) an, sodass er ohne Verzögerung weiterarbeiten kann. In dieser Arbeit wird die mächtigere Sichtweise 2 verwendet, d.h. ein mobiler Endanwender kann sich auch an mehr als einem Smart-Mobilgerät anmelden.

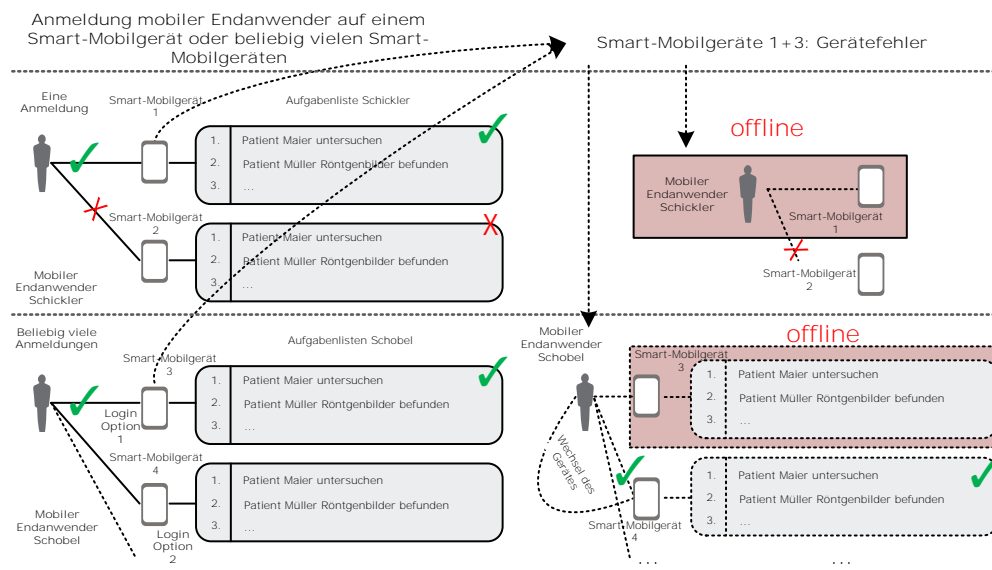


Abbildung 7.3: Verknüpfungsarten mobiler Endanwender mit Smart-Mobilgeräten

Die vier identifizierten Kategorien und ihre 18 Parameter bilden das Fundament des Parameterkatalogs. Gemeinsam mit dessen Regeln und Sonderfällen entsteht der mobile Kontext (siehe Abb. 7.2④). Letzterer bildet die Basis für eine kontextbezogene mobile Endanwenderzuteilung und die mobile Ausnahmebehandlungsstrategie, sodass die Ausführung mobiler Aktivitäten robust und kontextbezogen realisiert werden kann. In Abbildung 7.2④ wird unten ein mobiler Kontext für ein Prozessfragment illustriert.

7.3 Parameterkatalog

Tabelle 7.1 fasst die 18 Parameter des Parameterkatalogs zusammen, die für die kontextbezogene mobile Endanwenderzuteilung und die mobile Ausnahmebehandlungsstrategie genutzt werden. Tabelle 7.1 zeigt ferner, welche der Parameter funktionale Abhängigkeiten aufweisen. Wie Letztere aufgelöst werden, wird im Kontext der Modellierung und Ausführung mobiler Aktivitäten diskutiert. Ferner zeigt Tabelle 7.1, ob ein Parameter von der kontextbezogenen mobilen Endanwenderzuteilung oder der mobilen Ausnahmebehandlungsstrategie verwendet wird. Schließlich wird in der letzten Spalte gezeigt, ob es Sonderfälle zu betrachten gilt.

7.3.1 Parameterkategorie I: Smart-Mobilgerät

Die Parameter der Kategorie I beziehen sich auf das Smart-Mobilgerät.

Parameter 1 (Batterieladestatus): Bei einem Smart-Mobilgerät ist der Ladestatus der Batterie zu berücksichtigen. Wird einem mobilen Endanwender eine mobile Aktivität zur Ausführung übertragen, dessen Smart-Mobilgerät einen niedrigen Batterieladestatus hat, ist die Wahrscheinlichkeit hoch, dass es zu Ausnahmen oder Verzögerungen bei der Ausführung kommt. Im ungünstigsten Fall führt das dazu, dass die mobile Aktivität ohne Ladevorgang nicht beendet werden kann. Wenn nur dieser Endanwender die mobile Aktivität ausführen kann, würde es ihn zu einer Ladestation zwingen, was wiederum die Ausführungszeit der mobilen Aktivität verzögert. Im Extremfall fährt der mobile Endanwender sein Smart-Mobilgerät sogar zum Laden herunter, was wiederum schwer von einer abgebrochenen Konnektivität zu unterscheiden wäre. Aus diesen Gründen werden die für eine mobile Aktivität qualifizierten mobilen Endanwender, die aber einen niedrigen Ladestatus der Batterie ihres Smart-Mobilgeräts vorweisen, bei der Zuteilung zu mobilen Aktivitäten speziell berücksichtigt. Um den Ladestatus der Batterie in die Aktivitätenausführung einzubringen, wird der Parameter SMG_{BLS} (BLS: Batterieladestatus) eingeführt. Dieser wird mit dem korrespondierenden Parameter zum notwendigen Batterieladestatus einer mobilen Aktivität verglichen.

Parameter 2 (Formfaktor): In Kapitel 2 wurde gezeigt, dass der Formfaktor von Smart-Mobilgeräten [For14] für die Entwicklung mobiler Anwendungen wichtig ist. Unter Formfaktor versteht man die Einteilung von Smart-Mobilgeräten in Größen- und Funktionskategorien, etwa ob es sich um ein Smartphone [Sma14] oder einen Tablet-Computer [Tab14] handelt. In der Literatur wird noch eine weitere Kategorie genannt, die Phablets oder Smartlets [BW13]. Diese stellen von der Größe und Funktionsvielfalt gesehen einen Formfaktor zwischen Smartphone und Tablet-Computer dar. Weitere Kategorien sind in Zukunft zu erwarten. Im Kontext dieser Arbeit genügt es, die Formfaktoren Smartphone und Tablet-Computer zu betrachten. Dabei sind folgende Unterschiede relevant:

- **Display-Größe:** Smartphones haben im Vergleich zu Tablet-Computern ein kleineres Display. Die Display-Größe ist für die Oberflächengestaltung einer mobilen Anwendung wichtig. Meistens entscheidet man sich bei der Entwicklung der Oberfläche nur für eine Formfaktor-Kategorie, was zum Beispiel der Usability der Anwendung nicht immer zuträglich ist. Generell sollte die Zuteilung der mobilen Aktivität den Formfaktor des ausführenden Smart-Mobilgeräts berücksichtigen, sofern dies möglich ist. Dazu wird der Parameter SMG_{FF} (FF: Formfaktor) eingeführt. Ferner existiert wiederum ein korrespondierender Parameter für mobile Aktivitäten, der mit diesem verglichen wird.

PM	Beschreibung	Art	PA	ABSM	KMEZ	SV
Parameterkategorie I: Smart-Mobilgerät (SMG)						
SMG_{BLS}	Batterieladestatus	Messwert	X	✓	✓	X
SMG_{FF}	Formfaktor	Symbolwert	X	✓	✓	(✓)
SMG_{ONT}	Online-Netzwerktyp	Messwert	$SMG_{ONT} \rightarrow SMG_{BLS}$	✓	✓	X
SMG_{GAO}	Geometrischer Aufenthaltsort	Messwert	X	✓	✓	(✓)
Parameterkategorie II: Mobile Aktivität (MA)						
MA_{SAO}	Symbolische Ausführungsorte	Symbolwert	X	✓	✓	(✓)
MA_{GAO}	Geometrische(r) Ausführungsort(e)	Messwert	$MA_{GAO} \rightarrow MA_{EB}$	✓	✓	(✓)
MA_{EB}	Einzugsbereich	Symbolwert (Konstante)	X	✓	✓	(✓)
MA_{NBLS}	Notwendiger Batterieladestatus	Symbolwert (Konstante)	$MA_{NBLS} \rightarrow MA_{SFOS}$	✓	✓	(✓)
MA_D	Dringlichkeit	Symbolwert (Konstante)	$MA_D \rightarrow MA_{SFOS}$	✓	✓	X
MA_{OFF}	Offline-Ausführungsmöglichkeit	Symbolwert (Konstante)	$MA_{OFF} \rightarrow MA_D$ $MA_{OFF} \rightarrow MA_{SFOS}$	✓	✓	✓
MA_{GFF}	Geeigneter Formfaktor	Symbolwert	X	✓	✓	(✓)
MA_{SFOS}	Sendefrequenz zur Ermittlung des Online-Status	Symbolwert (Konstante)	X	✓	X	X
MA_{AZOS}	Abfragezeitpunkte zur Ermittlung des Online-Status	Symbolwert	X	✓	X	X
MA_{SWME}	Schwellwert Anzahl notwendiger mobiler Endanwender	Symbolwert (Konstante)	X	X	✓	X
Parameterkategorie III: Gesamtprozess (GP)						
GP_{DA}	Angabe zu Datenabhängigkeiten	Messwert	$GP_{DA} \rightarrow MA_{SFOS}$	✓	✓	X
GP_{ISZ}	Instant-Shutdown Referenzwert	Symbolwert		✓	✓	X
Parameterkategorie IV: Mobiler Endanwender (ME)						
ME_{SAO}	Symbolische Aufenthaltsorte	Symbolwert	X	✓	✓	(✓)
ME_{ISZ}	Instant-Shutdown Zähler	Messwert	X	✓	✓	X

$PM=Parameter$, $PA=Parameterabhängigkeit$
 $ABSM=Ausnahmebehandlungsstrategie mobiler Aktivitäten$,
 $KMEZ=Kontextbezogene mobile Endanwenderzuteilung$, $SV=Sonderfälle vorhanden$,
 $\checkmark=trifft zu$, $X=trifft nicht zu$,
 $(\checkmark)=Spezieller Sonderfall$, wenn Parameter nicht explizit gesetzt wurde,
 $\rightarrow = A \rightarrow B : B \text{ ist funktional abhängig von } A$

Tabelle 7.1: Parameter-Übersicht

- **Integrierte Sensoren:** Smartphones haben oftmals mehr integrierte Sensoren als Tablet-Computer. So ist in nahezu jedem Smartphone ein GPS-Sensor verbaut, was bei Tablet-

Computern meistens nicht der Fall ist. Auch eine Anbindung an das Telefonnetz bleibt häufig Smartphones vorbehalten und muss bei Tablet-Computern zugekauft werden. Die unterschiedliche Sensorik von Smart-Mobilgeräten verschiedener Formfaktoren ist eine wichtige Größe. Eine Erfassung von Sensoren erscheint daher sinnvoll. Aktuell werden in unserem Ansatz Sensoren aber nicht durch einen Parameter erfasst, da für jeden Sensor weitere Informationen (z.B. verschiedene Protokollausprägungen) verwaltet werden müssen. Deren Realisierung würde den Umfang der vorliegenden Arbeit aber sprengen.

Parameter 3 (Online-Netzwerktyp): Der Netzwerktyp, über den ein Smart-Mobilgerät Online geht, ist ebenfalls eine wichtige Größe. Entweder wird die Verbindung über ein Wireless LAN (WLAN) oder ein Telefonnetz hergestellt. Diese beide Arten der Verbindung unterscheiden sich in mehreren Punkten. Eine WLAN-Verbindung ist im Regelfall um ein Vielfaches rascher, stabiler und verbraucht weniger Batterieladestatus [PHZ12]. Generell verbraucht ein WLAN die gleiche Batterieladestatus wie eine durch ein Telefonnetz hergestellte Verbindung. Das energieeffiziente Element sind Verbindungsabbrüche, da ein erneuter Verbindungsaufbau viel Strom verbraucht [PHZ12]. Da Telefonnetzverbindungen generell instabiler sind sowie oft zwischen verschiedenen Arten des Telefonnetztyps, etwa UMTS [UMT14] zu EDGE [EDG14] (und umgekehrt), umgeschaltet wird, wird entsprechend mehr Batterieladestatus beansprucht als bei einem WLAN. Auch dass bei einer WLAN-Verbindung Daten rascher übertragen werden spart Batterieladestatus. Ferner muss bei einer WLAN-Verbindung nicht auf ein Kontingent (Mehrpreis ab einem festgelegten verbrauchten Datenvolumen) Rücksicht genommen werden. Um dies abzubilden, wird der Parameter SMG_{ONT} (ONT: Online-Netzwerktyp) eingeführt; er wird zur Laufzeit einer mobilen Aktivität mehrfach ausgewertet. Erstens wird der Batterieladestatus SMG_{NBS} anders gewichtet, wenn das Smart-Mobilgerät im Telefonnetz angemeldet ist. Zweitens werden zwischen den verschiedenen Netztypen Präzedenzen festgelegt, etwa dass ein mobiler Endanwender im WLAN einem mobilen Endanwender im Telefonnetz vorgezogen wird. Drittens wird für ein Smart-Mobilgerät mit Telefonnetzverbindung eine andere Online-Statusermittlung durchgeführt als für Geräte mit WLAN-Verbindung. Für Letztere wird der Status in kleineren Abständen ermittelt, da davon auszugehen ist, dass Telefonnetzverbindung öfter abbrechen können und dann meist in sehr kurzer Zeitspanne wieder aufgebaut werden. Aus diesem Grund vermeiden wir es durch Wahl längerer Abfrageintervalle, jeden kurzen Offline-Status bewerten zu müssen.

Parameter 4 (Geometrisch ermittelter Aufenthaltsort): Der Parameter SMG_{GAO} speichert den geometrisch ermittelten Aufenthaltsort eines Smart-Mobilgeräts. Auch für mobile Aktivitäten wird ein korrespondierender Parameter für den geometrisch ermittelten Aufenthaltsort eingeführt, um besonders geeignete Ausführungspaare zu identifizieren. Generell ist die Ermittlung geometrisch ermittelter Koordinaten nicht einfach. Das gilt sowohl für den Außen- als auch Innenbereich. Im Außenbereich nutzt man zum Beispiel das GPS-System [GPS14]. Fährt man aber durch einen Tunnel, fällt die Bestimmbarkeit des Ortes aus, da kein Satellitenempfang mehr besteht. Eine derartige Positionsbestimmung kann nicht als sicher angesehen werden. Im Innenbereich (d.h. in Gebäuden) findet man Lösungen wie das *Active Badge System* [WJH97], das wiederum mit hohem Aufwand für Installation und Wartung verbunden ist. Ferner fällt auch bei diesem Verfahren die Bestimmbarkeit oftmals aus [Jan07]. Dennoch ist ein Vergleich auf Basis geometrischer ermittelter Daten die exakteste Möglichkeit.

7.3.2 Parameterkategorie II: Mobile Aktivität

Die Parameter der Kategorie II beziehen sich auf die mobile Aktivität.

Parameter 5 und 6 (Symbolischer Ausführungsort und geometrisch ermittelter Ausführungsort): Bei mobilen Aktivitäten liegt es nahe, den Ort ihrer Ausführung zu berücksichtigen. So macht es einen Unterschied, ob eine mobile Aktivität außerhalb oder innerhalb eines Gebäudes ausgeführt werden soll. Im ersten Fall kann nicht von einer raschen und stabilen Netzwerkverbindung ausgegangen werden, wohingegen im zweiten Fall eine bessere Verbindung angenommen werden darf. Wenn man einerseits den Ort der mobilen Aktivität spezifiziert und andererseits den Aufenthaltsort eines mobilen Endanwenders bestimmen kann, lässt sich auch ermitteln, welche mobilen Endanwender aufgrund ihrer aktuellen Lokation besonders zur Ausführung einer mobilen Aktivität geeignet sind. Das sind genau diejenigen, die sich am Ort der mobilen Aktivität befinden (vgl. Abb. 7.4).

Im Beispiel aus Abbildung 7.4 soll durch die mobile Aktivität einem Patienten ein zentraler Zugang gelegt werden, d.h. ein Zugang direkt in eine der großen Körper-Venen. Damit lassen sich Medikamente rasch und in größeren Mengen in den Blutkreislauf einbringen. Diese Maßnahme birgt jedoch Risiken, weshalb sie nur in speziellen Räumen einer Klinik durchgeführt wird. Wie Abbildung 7.4 zeigt, kommen drei Raumarten infrage: Aufwachraum, Schockraum und Unfall-OP. Nun ist die Ausführung der mobilen Aktivität vor allem für diejenigen mobilen Endanwender geeignet, die sich in der Nähe oder exakt in einem der drei Räume aufhalten.

Zu diesem Zweck wird der Parameter MA_{SAO} (SAO: Symbolischer Ausführungsort) eingeführt. Er gibt an, an welchem Ort die Aktivität ausgeführt werden soll. Für den Parameter können ferner mehrere Orte angegeben werden (vgl. Abb. 7.4). Weiters gibt es Szenarien, in denen man den Ort der Ausführung nicht angeben kann oder eventuell auch nicht will. Daher ist es zulässig, dass MA_{SAO} aus einer leeren Menge besteht.

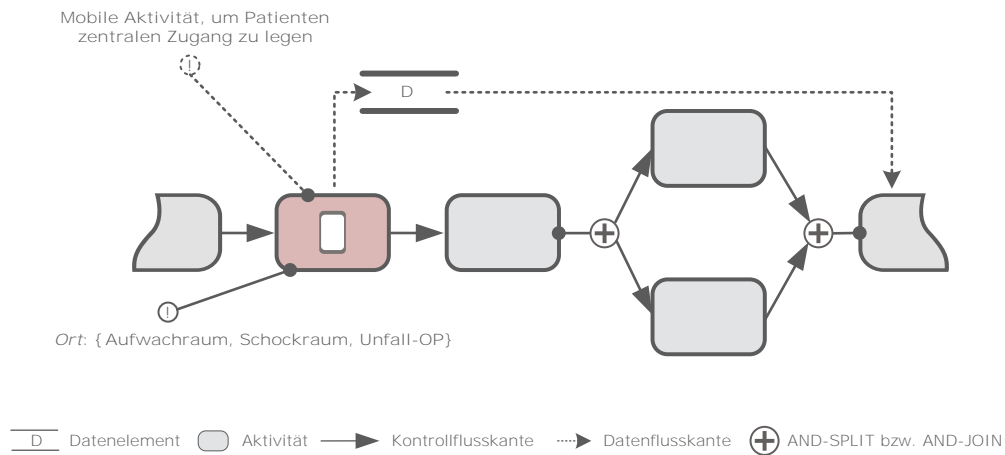


Abbildung 7.4: Prozessfragment mit festgelegtem Ausführungsort für mobile Aktivität

Aus den genannten Gründen ist es sinnvoll, den Ausführungsort einer mobilen Aktivität zu verwalten. Ein weiterer wichtiger Aspekt besteht darin, ob man den Ausführungsort symbolisch definiert oder ein Koordinatensystem zugrunde legt (d.h. den Ort geometrisch festlegt). In [BD05] findet man eine ausführliche Diskussion zu diesem Thema. Eine weitere Vertiefung

findet sich in [HA95] und [Dec11]. Wir verwenden eine Kombination aus den beiden genannten Methoden. Sowohl die eine als auch die andere Methode hat ihre Stärken bzw. Schwächen, so dass eine kombinierte Anwendung den größten Nutzen verspricht.

Symbolische Koordinaten sind einfach zu modellieren, aber weniger exakt bei der Anwendung. Man kann die symbolischen Koordinaten eines mobilen Endanwenders und einer mobilen Aktivität rasch und effizient vergleichen, hat dann aber keinen Anhaltspunkt, ob sich der mobile Endanwender auch wirklich bei der mobilen Aktivität befindet. Geometrisch ermittelte Koordinaten sind hingegen schwieriger zu ermitteln (siehe Diskussion zu SMG_{GAO}). Dennoch sind sie für eine Zuteilung besser geeignet; es muss aber einkalkuliert werden, dass diese nicht immer bestimmbar sind. Aus diesen Gründen werden für den Ausführungsort einer mobilen Aktivität zwei Parameter eingeführt: (1) MA_{SAO} legt die symbolischen und (2) MA_{GAO} die geometrischen Koordinaten fest.

Parameter 7 (Einzugsbereich des Ausführungsorts): Es wird für eine mobile Aktivität ein Einzugsbereich eingeführt (vgl. Abb. 7.5). Er besteht aus einem Bereich um den Ausführungsort der mobilen Aktivität. Weiters zeigt Abbildung 7.5, wann ein mobiler Endanwender geeignet ist (*Treffer*), die mobile Aktivität auszuführen. Dies ist genau dann der Fall, wenn er sich innerhalb des Einzugsbereichs aufhält. Dieser anwendungsabhängige Bereich wird für jede mobile Aktivität festgelegt und im Parameter MA_{EB} gespeichert. Innerhalb des Einzugsbereichs werden alle mobilen Endanwender gleich behandelt. So ist der mobile Endanwender mit *Treffer 1* aus Abbildung 7.5 nicht besser geeignet, um die mobile Aktivität auszuführen, als der mobile Endanwender mit *Treffer 2*.

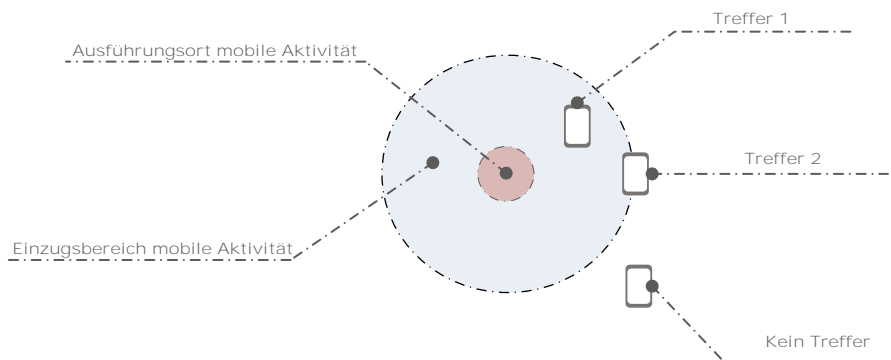


Abbildung 7.5: Einzugsbereich und Ausführungsort einer mobilen Aktivität

Parameter 8 (Notwendiger Batterieladestatus): Für mobile Aktivitäten wird der Parameter MA_{NBLS} (NBLS: Notwendiger Batterieladestatus) eingeführt. Er gibt an, wie hoch der Batterieladestatus eines Smart-Mobilgeräts mindestens sein soll, damit sein Endanwender eine bestimmte mobile Aktivität auf diesem Gerät ausführen darf.

Parameter 9 (Dringlichkeit): Speziell in mobilen Umgebungen, in denen sich Bedingungen rasch ändern können, ist man oft darauf angewiesen, dass eine Aktivität zu einem bestimmten Zeitpunkt erledigt sein muss. Nachfolgende Aktivitäten hängen davon ab, dass die vorhergehende Aktivität termingerecht oder innerhalb einer bestimmten Zeitdauer erledigt wird. Zur

Veranschaulichung soll ein Beispiel dienen: Ein Notfallpatient wird in eine Klinik eingeliefert und bereits in der Aufnahme wird klar, dass der Patient notoperiert werden muss. Vorbereitend für die Notoperation ist ein Labortest notwendig, der die Blutgerinnung bestimmt. Da es sich um eine Notfall-OP handelt, muss der Labortest zu einem bestimmten Zeitpunkt beendet sein. Um solche *Dringlichkeiten* mobiler Aktivitäten zu berücksichtigen, wird der Parameter MA_D eingeführt. Er gibt entweder einen konkreten Zeitpunkt (z.B. 15:00 Uhr) oder eine relative Zeitdauer (z.B. 5 Stunden) an.

Parameter 10 (Offline-Aktivität): Es gibt mobile Umgebungen, in denen ein mobiler Endanwender eine mobile Aktivität ausführt, aber während der Ausführung dann Offline gehen muss. Zur Veranschaulichung sollen zwei Beispiele dienen:

1. Ein Außendienstmitarbeiter sei in einen Prozess eingebunden, der die Wartung einer Maschine beim Kunden überwacht. Weiters sei der Außendienstmitarbeiter für diejenige Aktivität zuständig, die beim Kunden vor Ort auszuführen ist. Kann nun keine Online-Verbindung beim Kunden aufgebaut werden, sollte der Außendienstmitarbeiter dennoch weiterarbeiten können, ohne dass das PrMS eine Ausnahmesituation erkennt.
2. Ein weiteres Beispiel sind Räume in einer Klinik, in denen keine Verbindung erlaubt ist, etwa damit eine Netzwerkstrahlung andere Geräte nicht beeinträchtigt. Auch in solchen Fällen ist eine fehlende Konnektivität keine Ausnahmesituation. Um Offline-Aktivitäten abzubilden, wird der Parameter MA_{OFF} eingeführt. Er gibt für eine mobile Aktivität an, dass die mobile Aktivität auch Offline ausgeführt werden darf. Wie lange die Offline-Situation toleriert wird, wird über die funktionale Abhängigkeit zwischen den Parametern MA_{OFF} und MA_D geregelt (vgl. Abb. 7.6). MA_D gibt an, ab wann die Offline-Situation nicht mehr toleriert wird, und die mobile Ausnahmebehandlungsstrategie gestartet wird. Außerdem wird bei gesetztem Parameter MA_{OFF} der Parameter MA_{SFOS} nicht ausgewertet. Das heißt, MA_{SFOS} ist funktional abhängig von MA_{OFF} .

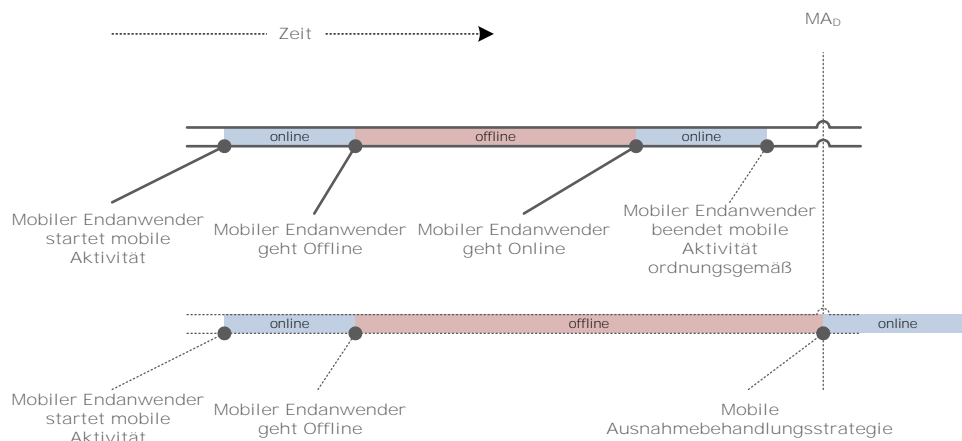


Abbildung 7.6: Funktionale Abhängigkeit von MA_{OFF} und MA_D

Da Offline-Aktivitäten im Kontext mobiler Aktivitäten sehr wichtig sind, wird deren Unterstützung kurz zusammengefasst:

Mobile Offline-Aktivitäten

Mobile Offline-Aktivitäten werden explizit unterstützt. Mobile Endanwender können dazu eine festgelegte Zeitdauer Offline bleiben, ohne dass die Ausnahmebehandlung ausgeführt wird. Damit wird die Ausführungsmächtigkeit mobiler Aktivitäten deutlich erhöht.

Parameter 11 (Geeigneter Formfaktor): Als Antagonist des Parameters SMG_{FF} existiert der Parameter MA_{GFF} (GFF: Geeigneter Formfaktor). Er bestimmt, welche Formfaktoren von Smart-Mobilgeräten geeignet sind, eine konkrete mobile Aktivität auszuführen. Er enthält entweder den Wert *null* oder die für die mobile Aktivität geeigneten Formfaktoren.

Parameter 12 und 13 (Parameter für Online-Status Prüfung): Für die Ausführung einer mobilen Aktivität ist es grundlegend, ob ein mobiler Endanwender mit seinem Smart-Mobilgerät tatsächlich *Online* ist. Nur dann kann ihm die Ausführung der mobilen Aktivität übertragen werden. Ferner muss er während ihrer Ausführung auch Online sein, da sonst keine Informationen mit dem *PrMS* ausgetauscht werden können. Online zu sein bedeutet, dass der mobile Endanwender in einem Netzwerk angemeldet ist, über das ihn das *PrMS* kontaktieren kann. Weitere mobile Endanwender, die zwar berechtigt sind, die mobile Aktivität auszuführen, die aber nicht Online sind, werden dementsprechend nicht berücksichtigt.

Generell sind der Online-Status sowie die Art seiner Ermittlung auch für die Ausnahmebehandlung mobiler Aktivitäten wichtig. Für die Ermittlung des Online-Status sind zwei Aspekte relevant: (1) Welche Abfragemethoden kommen zur Online-Statusermittlung grundsätzlich infrage? (2) Welche Komponente ermittelt den Online-Status? Tabelle 7.2 gibt die Antwort auf den ersten Aspekt. Was die zweite Fragestellung anbetrifft, gibt es drei Möglichkeiten (vgl. Tabelle 7.3).

Hiervon wird folgende Online-Statusermittlung in dieser Arbeit verwendet:

Angewandte Online-Statusermittlung

Der Parameter MA_{SFOS} gibt an, wie häufig die Online-Statusermittlung erfolgen soll. Ferner legt der Parameter MA_{AZOS} spezifische Zeitpunkte fest, an denen das Smart-Mobilgerät seines Status an das *PrMS* übermitteln soll. Beide Ermittlungsvarianten beziehen sich auf die Ausführungszeit einer mobilen Aktivität.³

Parameter 14 (Schwellwert): Ein Aspekt, der nicht nur für die Zuteilung mobiler Endanwendern zu mobilen Aktivitäten wichtig ist, sondern auch für die Ausnahmebehandlung, ist die Anzahl mobiler Endanwender, die für die Ausführung einer mobilen Aktivität geeignet sind und die auch zur Verfügung stehen.

Zur Veranschaulichung soll ein Beispiel dienen: Schreibt eine mobile Aktivität Daten und steht zur Ausführung nur ein geeigneter mobiler Endanwender zur Verfügung, würde bei Zuteilung der Aktivität an diesen Endanwender und einer darauffolgenden Ausnahme seines Smart-Mobilgeräts (z.B. Crash) eine erhebliche Verzögerung der Ausführung resultieren. Diese Verzögerung infolge fehlender mobiler Endanwender soll durch Einführung des Parameters MA_{SWME}

³SFOS: Sendefrequenz Online-Status; AZOS: Abfragezeitpunkte Online-Status

Online-Status Ermittlung Grundsätzliche Abfragemethoden	Beschreibung
<i>Online-Status wird kontinuierlich ermittelt.</i>	So stellt sich die Frage, was geeignete zeitliche Abstände zur Abfrage des Online-Status sind. Je robuster die Ausführung einer mobilen Aktivität sein soll, desto öfter sollte dieser Status ermittelt werden. Als Gegenargument kann angeführt werden, dass geringe Abstände der Abfragen dazu führen, dass unnötiger Netzwerkverkehr erzeugt wird. Da es sich aber nur um eine sehr kleine zu übertragene Datenmenge handelt, kann dieser Aspekt vernachlässigt werden. Weiters sollte ein Parameter existieren, mit dem sich dynamisch steuern lässt, wie groß die Abstände sind.
<i>Online-Status wird selektiv ermittelt.</i>	Die selektive Ermittlung des Online-Status gestaltet sich schwieriger als die kontinuierliche Ermittlung. Generell kann man zwei Arten unterscheiden. <i>Erstens</i> kann der Online-Status bei Bedarf ermittelt werden, etwa wenn eine mobile Aktivität einem mobilen Endanwender zugewiesen werden soll. <i>Zweitens</i> kann der Status bei bestimmten Ereignissen selektiv ermittelt werden, etwa wenn sich am System (neue) mobile Endanwender anmelden. Die zweite Variante ist für eine robuste Ausführung jedoch ungeeignet.

Tabelle 7.2: Methoden der Online-Status Ermittlung eines mobilen Endanwenders

(SWME: Schwellwert mobiler Endanwender) berücksichtigt werden. Dieser Parameter gibt an, wie viele mobile Endanwender zur Verfügung stehen sollten, damit die mobile Aktivität robust ausgeführt werden kann.

7.3.3 Parameterkategorie III: Gesamtprozess

Die Parameter der Kategorie III beziehen sich auf den Gesamtprozess.

Parameter 15 (Datenabhängigkeiten): Generell sind mobile Aktivitäten kritisch zu betrachten, wenn es um die robuste Ausführbarkeit eines Prozesses geht. Im Vergleich zu Aktivitäten, die nicht mobil ausgeführt werden, gestalten sich Ausnahmebehandlungen entsprechend schwieriger. Generell sollten Ausnahmen bei der Ausführung mobiler Aktivitäten rasch behoben werden (z.B. wegen sinkendem Batterieladestatus). Dies gilt in besonderem Maße für mobile Aktivitäten, die Daten schreiben. Daher wird der Parameter GP_{DA} (DA: Datenabhängigkeit) eingeführt. Er angibt an, ob eine mobile Aktivität Datenabhängigkeiten zu weiteren Aktivitäten hat. Falls ja, wird dies spezifisch von der kontextbezogenen mobilen Endanwenderzuteilung und der mobilen Ausnahmebehandlungsstrategie berücksichtigt.

Parameter 16 (Referenzwert Instant-Shutdowns): Der Parameter GP_{ISZ} wird für einen Prozess manuell festgelegt. Er gibt an, was die Toleranzgrenze für einen mobilen Endanwender in Bezug auf die Anzahl an Instant-Shutdowns darstellt. Überschreitet ein mobiler Endanwender diese Toleranzgrenze, wird er bei der kontextbezogenen mobilen Endanwenderzuteilung und der mobilen Ausnahmebehandlungsstrategie anders behandelt. Dieser Parameter wird mit dem korrespondierenden Parameter ME_{ISZ} verglichen.

Online-Statusermittlung			Beschreibung
Verantwortliche Komponente			
1	<i>PrMS</i>	Das <i>PrMS</i> prüft in definierten Abständen, ob ein mobiler Endanwender Online ist. D.h. das <i>PrMS</i> übernimmt die aktive Rolle zur Überprüfung des Online-Status.	Sowohl eine kontinuierliche als auch selektive Abfrage des Online-Status ist mit dieser Methode kombinierbar.
2	<i>Smart-Mobilgerät</i>	Das Smart-Mobilgerät schickt in definierten Abständen seinen Online-Status zum <i>PrMS</i> . Hier hat das <i>PrMS</i> nur eine passive Rolle bei der Überprüfung des Online-Status inne.	Da ein Smart-Mobilgerät nur wenige Ereignisse anderer mobiler Endanwender abfragen kann, erscheint bei dieser Methode nur eine kontinuierliche Abfrage des Online-Status sinnvoll. Jedoch sind einige Zustände und Ereignisse des Smart-Mobilgeräts während der Ausführung einer mobilen Aktivität interessant, etwa wenn die mobile Anwendung, die zur Ausführung der mobilen Aktivität genutzt wird, in den Hintergrund verdrängt wird.
3	<i>Hybrid</i>	Es wird eine Kombination der Methoden 1 und 2 verfolgt.	Hier gibt es verschiedene Kombinationsmöglichkeiten. Auf Seiten des Smart-Mobilgeräts ist sowohl die kontinuierliche als auch selektive Abfrage möglich. Außerdem kann das <i>PrMS</i> zu spezifischen Zeitpunkten zusätzlich den Online-Status ermitteln. Weiters ist denkbar, dass sowohl das Smart-Mobilgerät als auch das <i>PrMS</i> kontinuierlich den Online-Status ermitteln.

Tabelle 7.3: Ermittlungsvarianten des Online-Status mobiler Endanwender

7.3.4 Parameterkategorie IV: Mobiler Endanwender

Die Parameter der Kategorie IV beziehen sich auf den mobilen Endanwender.

Parameter 17 (Aufenthaltort eines mobilen Endanwenders): Zu einer mobilen Aktivität wird der Ausführungsort sowohl mittels geometrischer als auch symbolischer Koordinaten repräsentiert. Zu einem mobilen Endanwender wiederum werden nur symbolische Koordinaten für den Aufenthaltort gespeichert. Die geometrisch ermittelten Koordinaten ergeben sich durch sein Smart-Mobilgerät (vgl. Parameter 4), während die symbolischen Koordinaten im Parameter *ME_{SAO}* (SAO: Symbolischer Aufenthaltort) erfasst werden, für deren Verwaltung zwei Aspekte relevant sind:

1. Welchen Sinn ergibt eine symbolische Koordinate für mobile Endanwender, wenn sich diese im Prinzip beliebig bewegen können?
Die folgenden beiden Gründe sprechen für die Verwendung symbolischer Koordinaten: (1) Lassen sich in einer Umgebung aus technischen Gründen keine geometrischen Koordinaten zu einem mobilen Endanwender bestimmen, kann ein rascher Vergleich zwischen den symbolischen Koordinaten mobiler Endanwender und mobiler Aktivitäten erfolgen. Das heißt nicht, dass sich der mobile Endanwender bei einer Übereinstimmung am Ausführungsort befinden muss. Man weiß aber zumindest, dass dieser mobile Endanwender bevorzugt an dem Ort zu finden ist. Es kann natürlich vorkommen, dass ein anderer mobiler Endanwender eventuell näher ist, obwohl er den Ausführungsort nicht in seinen

symbolischen Koordinaten hat. Dieser Nachteil kann aber in Kauf genommen werden, wenn kein anderer Vergleich bezüglich des Ortes möglich ist.

2. Kann es mehrere symbolische Koordinaten für einen mobilen Endanwender geben?

Diese Frage ist ebenfalls zu bejahen, d.h. ein mobiler Endanwender kann mehrere bevorzugte Aufenthaltsorte haben. Man stelle sich einen Stationsarzt vor, der zeitweise auch in der Notaufnahme arbeitet.

Parameter 18 (Instant-Shutdown Verhalten): Eine Handlung, die sich bei mobilen Endanwendern oftmals beobachten lässt, sind *Instant-Shutdowns*. Ein Instant-Shutdown (1) wird meist spontan und ohne Berücksichtigung des aktuellen Ausführungszustands durchgeführt, (2) und er erfolgt oftmals im Kontext eines Ad-hoc-Zwischenfalls (d.h. einem notwendigen, aber nicht vorhergesehenem Kontextwechsel). Man denke an ein Notfallszenario in einer Klinik, bei dem sich der Gesundheitszustand eines Patienten schlagartig ändert. Oftmals kann hier mit der regulären Tätigkeit nicht fortgefahren werden und das Smart-Mobilgerät wird (unbedacht) ausgeschaltet. Mobile Endanwender, die solche Instant-Shutdowns allerdings häufig durchführen, sind für die Ausführung mobiler Aktivitäten kritischer zu betrachten als Endanwender, die dieses Verhalten weniger oft zeigen. Das soll nicht heißen, dass dieses Verhalten generell unachtsam ist. Allerdings zeigen Instant-Shutdowns an, dass der betreffende mobile Endanwender häufige Kontextwechsel durchführen muss. Ähnlich wie beim Batterieladestatus behandeln wir mobile Endanwender, die ein solches Verhalten zeigen, speziell.

Um das Verhalten des mobilen Endanwenders zu erfassen, wird der Parameter ME_{ISZ} (ISZ: Instant-Shutdown Zähler) eingeführt. Dieser wird jedes Mal um eins erhöht, wenn der mobile Endanwender einen Instant-Shutdown durchführt. Um festzustellen, ob ein Instant-Shutdown erfolgt ist, wird folgende Prozedur angewandt: Hat ein mobiler Endanwender eine mobile Aktivität ausgeführt und kam es während der Ausführung zu Ausnahmen, welche die Nichterreichbarkeit des Smart-Mobilgeräts betreffen, werden die internen Statusvariablen des Smart-Mobilgeräts ausgewertet, um einen Instant-Shutdown zu erkennen. Diese Auswertungsmöglichkeit wurde auf den wichtigen Smart-Mobilgeräte Plattformen evaluiert.⁴ Dieser Parameter wird mit dem korrespondierenden Parameter GP_{ISZ} verglichen.

7.3.5 Sonderfälle

Tabelle 7.1 unterscheidet zwei Arten von Sonderfällen: (1) diejenigen, bei denen ein leerer Parameterwert Auswirkungen hat und (2) diejenigen, die nicht in diese Kategorie fallen. Erstgenannte werden bei der Modellierung, Ausführung und Ausnahmebehandlung mobiler Aktivitäten (siehe Kapitel 8) diskutiert. Letztgenannte stellen sich wie folgt dar:

1. Sonderfall zum Parameter MA_{OFF} : Der Parameter MA_D ist funktional abhängig von MA_{OFF} (vgl. Tabelle 7.1). Angenommen eine Offline-Aktivität wird zu einem Zeitpunkt gestartet, zu dem der für MA_D gesetzte Zeitpunkt bereits überschritten wurde. Dann wird beim Starten der mobilen Aktivität zugelassen, dass der Wert während der Ausführung noch angepasst werden kann. Ansonsten ist keine Werteänderung symbolischer Parameter zur Ausführungszeit möglich.

⁴iOS, Android und Windows Mobile bieten die Möglichkeit, einen Instant-Shutdown auszulösen. Bei Android wird zum Beispiel die folgende interne Systemvariable ausgelesen: `andro-id.intent.action.ACTION_SHUTDOWN`

2. Sonderfall zum Parameter ME_{SAO} : Angenommen ein mobiler Endanwender, für den keine symbolischen Koordinaten festgelegt worden sind und auch keine geometrischen Koordinaten ermittelt werden konnten, befindet sich näher am Ausführungsort als ein anderer mobiler Endanwender mit übereinstimmenden symbolischen Koordinaten. Dann werden andere Parameter stärker gewichtet als die Ortsübereinstimmung, obwohl diese im Regelfall die höchste Präzedenz hat.

7.4 Weitere Aspekte

Wir diskutieren weitere Aspekte zum mobilen Kontext. Die vorgestellten Parameter des mobilen Kontexts wurden in vier Kategorien eingeteilt: Smart-Mobilgerät, Mobile Aktivität, Gesamtprozess und mobiler Endanwender. Eine weitere Maßnahme, um spezifisch auf Szenarien einzugehen, ist eine Gewichtung für einzelne Kategorien einzuführen. So könnten die Parameter für das Smart-Mobilgerät stärker gewichtet werden als Parameter für die mobile Aktivität. Spielen beispielsweise Batterieladestatus und Aufenthaltsort, beides Parameter der Kategorie Smart-Mobilgerät, in einem Szenario eine große Rolle, könnte die Kategorie Smart-Mobilgerät stärker gewichtet werden als die übrigen drei Kategorien. Die Anforderungsanalyse (vgl. Kapitel 2) hat gezeigt, dass dies sinnvoll sein kann. Eine solche Gewichtung muss jedoch manuell auf der Grundlage von Erfahrungswerten erfolgen.

Die Daten zum mobilen Kontext müssen in einer *PrMS*-Umgebung geeignet verwaltet werden. Dazu wird die in Kapitel 6 vorgestellte Referenzarchitektur erweitert. Abbildung 7.7 zeigt wie sich diese Erweiterung darstellt. Die geänderten Stellen sind entsprechend mit *I-III* markiert. Bei der Markierung *I* ist für die Modellierungsumgebung der Referenzarchitektur eine Komponente hinzugekommen, um den mobilen Kontext zu modellieren. Dazu gehören die Definition des Parameterkatalogs (Symbolwerte) und die Festlegung der Vorgehensweise bei Sonderfällen. Die Evaluation bestehender Umgebungen (z.B. *AristaFlow* [RDRM⁺09a], *Intalio* [HSS08] und *ARIS* [RRHB09]) hat gezeigt, dass eine solche Integration möglich ist.

Markierung *II* zeigt eine Repository-Komponente für die Speicherung der anfallenden Daten des mobilen Kontexts. Darunter fallen alle Informationen, die sich während der Modellierung des mobilen Kontexts ergeben, ebenso wie die während der Ausführung ermittelten Messwerte. Die Kommunikation mit dieser neuen Komponente (siehe Markierung *III*) erfolgt über die zentrale Kommunikationsschnittstelle des *PrMS*.

Durch Analysen der Parameter kann ermittelt werden, in welcher Konstellation mobile Aktivitäten effizient, robust und endanwendergerecht ausgeführt werden. Hier sind insbesondere Process-Mining Techniken nützlich [LRW09b, LRW09a, LRW10, LRW11]. Aus diesem Grund werden die für die Ausführung einer mobilen Aktivität verwendeten Parameterwerte in der Prozess-Ausführungshistorie gespeichert werden.

7.5 Diskussion

Dieser Abschnitt diskutiert die Eignung des mobilen Kontexts und bewertet den Aufwand für die Verwaltung des Parameterkatalogs.

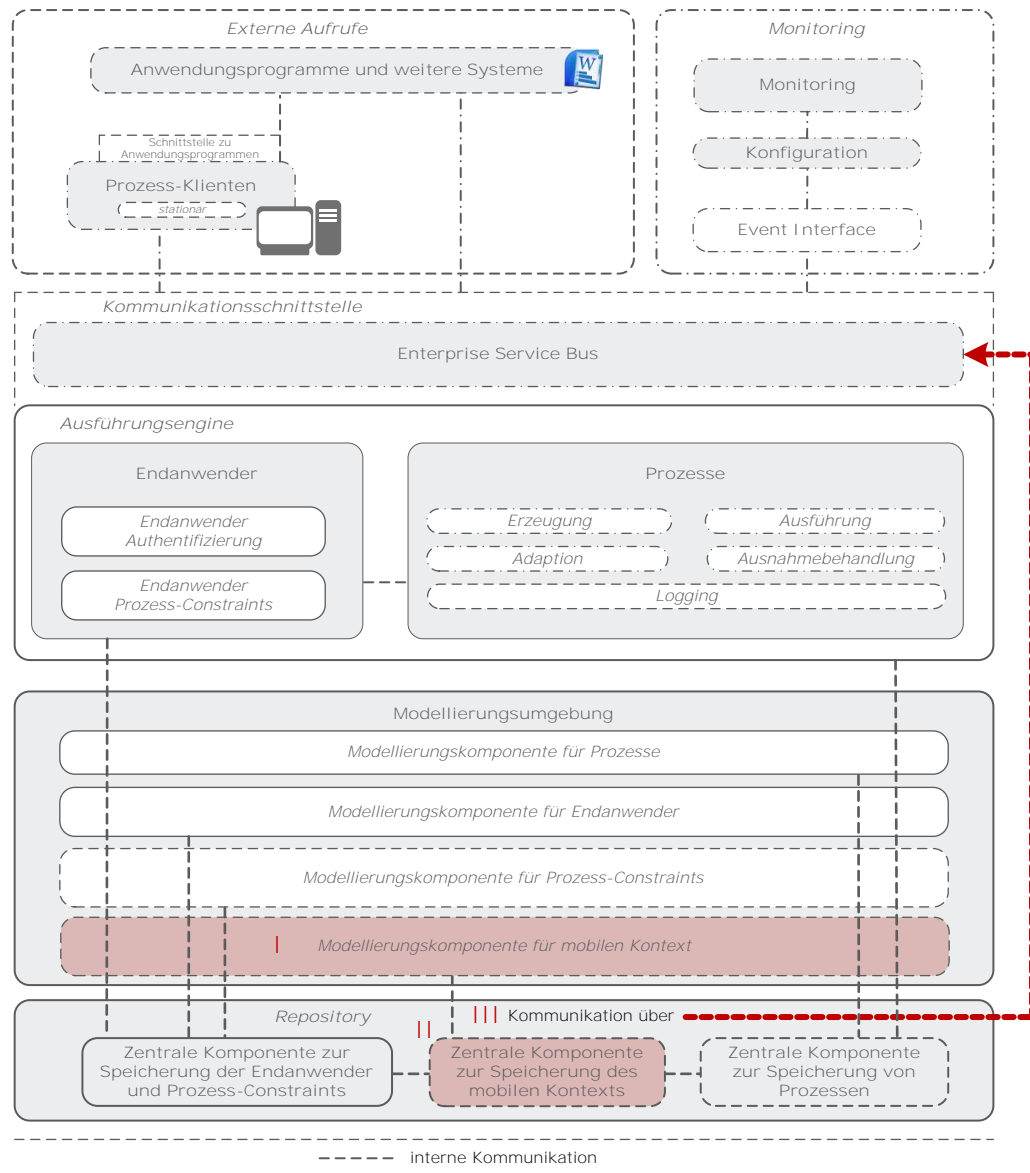


Abbildung 7.7: Modellierung und Verwaltung des mobilen Kontexts in der *PrMS*-Architektur

7.5.1 Eignung des mobilen Kontexts

Die kontextbezogene mobile Endanwenderzuteilung und die mobile Ausnahmebehandlungsstrategie verfolgen das Ziel, mobile Aktivitäten kontextbezogen und robust auszuführen. Die kontextbezogene mobile Endanwenderzuteilung zielt auf eine geeignete Zuteilung mobiler Endanwender zu mobilen Aktivitäten. Proaktiv sollen nur diejenigen mobilen Endanwender eine mobile Aktivität ausführen können, die mit geringerer Wahrscheinlichkeit ausfallen als andere. Da der Parameterkatalog unterschiedliche Kategorien von Parametern bietet, können unterschiedliche Aspekte des mobilen Kontexts für die Eignung eines mobilen Endanwenders herangezogen werden, etwa der Ort der Ausführung oder das Instant-Shutdown-Verhalten der

mobilen Endanwender. Beide Aspekte können unabhängig von der kontextbezogenen mobilen Endanwenderzuteilung ausgewertet werden. Aus den genannten Gründen kann hergeleitet werden, dass der mobile Kontext eine geeignete Basis für die kontextbezogene mobile Endanwenderzuteilung sowie die mobile Ausnahmebehandlungsstrategie darstellt.

Zwei Kategorien werden bislang für den mobilen Kontext ausgeklammert. Die erste bezieht sich auf die Sensorik von Smart-Mobilgeräten, die hier nicht weiters behandelt wird [SSP⁺13]. Die zweite Kategorie betrifft Endanwenderprofile [CFZ01], mittels deren zum Beispiel das Ausführungsprotokoll zwischen Prozess-Klient und *PrMS* noch besser berücksichtigt werden kann [FGK⁺05].

7.5.2 Spezifikations- und Verwaltungsaufwand

Ein ebenfalls wichtiger Aspekt betrifft die Wartbarkeit des Parameterkatalogs. Beispielsweise stellt sich die Frage, ob ein Parameter für jede Prozessinstanz neu spezifiziert werden muss oder global für einen Prozesstyp festgelegt werden kann. So sollte sich die Spezifikation eines Parameters für einen Endanwender nicht zu komplex darstellen.

Tabelle 7.4 beantwortet die Frage nach dem Spezifikations- und Verwaltungsaufwand des Parameterkatalogs. Die Tabelle zeigt ausschließlich Parameter, die nicht automatisch vom *PrMS* ermittelt werden.⁵

Es verbleiben 12 der 18 Parameter, die weiter zu diskutieren sind. Dazu werden die letzten drei Spalten aus Tabelle 7.4 behandelt. Die Spalte *PA* führt die funktionalen Parameterabhängigkeiten auf, für die eine manuelle Angabe erfolgen muss. So ist festzulegen, wie lange bei einer Offline-Aktivität die Offline-Situation zu keiner Ausnahmebehandlung führen soll. Eine solche Festlegung ist jedoch mit vertretbarem Aufwand möglich, da Erfahrungswerte herangezogen werden können.

Einen Parameter für jede Prozessinstanz neu festlegen zu müssen (vgl. Spalte *PPI*), würde einen hohen Spezifikations- und Verwaltungsaufwand bedeuten. Wie in Tabelle 7.4 ersichtlich wird, werden alle Parameter für Prozesstypen angegeben, das heißt zur Ausführungszeit entsteht kein zusätzlicher Aufwand. Dies trägt zur Reduktion des Aufwands für die Spezifikation und Verwaltung erheblich bei.

Abschließend soll geklärt werden, ob manuell festzulegende Parameter von Erfahrungswerten abhängen. Letztere beeinflussen die Güte (d.h. Optimierungsmöglichkeit) des Parameters. In Tabelle 7.4 sind alle Parameter, die durch Erfahrungswerte festgelegt werden, farblich markiert. Ein Beispiel für einen solchen Parameter ist *MANBLS*. Wird für mobile Aktivitäten ein zu hoher Batterieladestatus gefordert, qualifizieren sich ggf. keine mobilen Endanwender für die Ausführung einer mobilen Aktivität. Letztlich muss für Parameter, die von Erfahrungswerten abhängen, in Kauf genommen werden, dass deren festgelegter Wert eventuell verbessert werden könnte.

Wie Tabelle 7.4 zeigt, sind es 5 Parameter, für die Erfahrungswerte wichtig sind. Da einerseits nur der geringere Teil aller 18 Parameter von Erfahrungswerten abhängt und andererseits dennoch wichtige Informationen enthält, kann auch für diese der Aufwand für die Spezifikation und Verwaltung als vertretbar angesehen werden. Darüber hinaus stellen sich gute Erfahrungswerte meist rasch ein.

⁵Letztgenannte erzeugen einen geringen Modellier- und Verwaltungsaufwand und werden daher in Tabelle 7.4 ausgespart.

7.6 Zusammenfassung

Kapitel 7 hat den mobilen Kontext behandelt, dessen Basis ein umfassender Parameterkatalog bildet. Es konnte gezeigt werden, dass der Parameterkatalog ein leistungsfähiges Werkzeug für die kontextbezogene mobile Endanwenderzuteilung und die mobilen Ausnahmebehandlungsstrategie darstellt. Der Parameterkatalog bildet damit die Grundlage zur robusten und kontextbezogenen Ausführung mobiler Aktivitäten. Schließlich wurde gezeigt, dass der Parameterkatalog trotz seiner Komplexität bzw. Umfangs einen vertretbaren Spezifikations- und Verwaltungsaufwand für Endanwender erzeugt.

PM	Beschreibung	Art	Regel	PA	PPT	PPI
Parameterkategorie I: Smart-Mobilgerät (SMG)						
SMG_{FF}	Formfaktor	Symbolwert	X	X	✓	X
Parameterkategorie II: Mobile Aktivität (MA)						
MA_{SAO}	Symbolische Ausführungsorte	Symbolwert	X	X	✓	X
(MA_{EB})	Einzugsbereich	Symbolwert (Konstante)	X	X	✓	X
(MA_{NBLS})	Notwendiger Batterieladestatus	Symbolwert (Konstante)	$MA_{NBLS} \rightarrow MA_{SFOS}$	X	✓	X
MA_D	Dringlichkeit	Symbolwert (Konstante)	$MA_D \rightarrow MA_{SFOS}$	X	✓	X
MA_{OFF}	Offline-Ausführungsmöglichkeit	Symbolwert (Konstante)	$MA_{OFF} \rightarrow MA_D$	X	✓	X
MA_{GFF}	Geeigneter Formfaktor	Symbolwert	X	X	✓	X
(MA_{SFOS})	Sendefrequenz zur Ermittlung des Online-Status	Symbolwert (Konstante)	X	X	✓	X
(MA_{AZOS})	Abfragezeitpunkte zur Ermittlung des Online-Status	Symbolwert	X	X	✓	X
(MA_{SWME})	Schwellwert Anzahl notwendiger mobiler Endanwender	Symbolwert (Konstante)	X	X	✓	X
Parameterkategorie III: Gesamtprozess (GP)						
GP_{ISZ}	Instant-Shut down Referenzwert	Symbolwert	X	X	✓	X
Parameterkategorie IV: Mobiler Endanwender (ME)						
ME_{SAO}	Symbolische Aufenthaltsorte	Symbolwert	X	X	✓	X

PM =Parameter,
 PA =Parameterabhängigkeit mit manueller Festlegung,
 PPT =Für Prozesstyp definiert, PPI =Für jede Instanz eines Prozesstyps definiert
✓=trifft zu, X =trifft nicht zu,
 $\rightarrow = A \rightarrow B : B$ ist funktional abhängig von A

Tabelle 7.4: Parameter und ihr Spezifikations- und Verwaltungsaufwand

*Es gibt nur eine falsche Sicht der Dinge: der Glaube, meine Sicht
sei die einzig richtige.*

Nagarjuna (ca. 2. Jahrhundert)

8

Modellierung, Erzeugung und Ausführung mobiler Aktivitäten

Das vorliegende Kapitel befasst sich mit der Modellierung, Erzeugung und Ausführung mobiler Aktivitäten. Das Ziel ist die kontextbezogene Zuteilung mobiler Endanwender zu mobilen Aktivitäten. Wie bereits in Kapitel 7 diskutiert, sollen dadurch in proaktiver Weise nur diejenigen mobilen Endanwender zur Ausführung einer mobilen Aktivität ausgewählt werden, die mit einer geringeren Quote als nicht gewählte mobile Endanwender eine Ausnahme verursachen. Wie dies mithilfe des mobilen Kontexts erreicht werden kann, wird in diesem Kapitel gezeigt. Dazu wird ein Algorithmus vorgestellt, der ein *Ranking* der für eine mobile Aktivität prinzipiell infrage kommenden mobiler Endanwender berechnet. Dieser bildet die Basis für die kontextbezogene Zuteilung mobiler Endanwender zu mobilen Aktivitäten.

Kapitel 8 gliedert sich wie folgt: Abschnitt 8.1 motiviert die Notwendigkeit einer kontextbezogenen mobilen Endanwenderzuteilung im Kontext der Modellierung, Erzeugung und Ausführung mobiler Aktivitäten. Abschnitt 8.2 behandelt die Modellierung, Abschnitt 8.3 die Erzeugung und Abschnitt 8.4 schließlich die Ausführung mobiler Aktivitäten. Spezielle Aspekte dieser drei Phasen werden in Abschnitt 8.5 vorgestellt. Der Abschnitt 8.6 diskutiert, inwieweit die Ausführung mobiler Aktivitäten durch die vorgestellten Maßnahmen robuster wird und vergleicht den Ansatz mit verwandten Arbeiten. Abschnitt 8.7 fasst das Kapitel zusammen.

8.1 Einleitung

In Kapitel 5 wurde gezeigt, dass Prozesse, ebenso wie Aktivitäten, einem Lebenszyklus unterliegen. Der Lebenszyklus mobiler Aktivitäten (d.h. Modellierung, Erzeugung und Ausführung) unterscheidet sich jedoch von dem nicht mobiler Aktivitäten. Dies erfordert eine entsprechende Erweiterung des in Kapitel 5 vorgestellten Metamodells. Diese Erweiterungen werden notwendig, um die angestrebte kontextbezogene mobile Endanwenderzuteilung zu realisieren. Das erweiterte Metamodell wird in Abbildung 8.1 dargestellt.¹ Die Erweiterungen des Metamodells tragen einerseits der kontextbezogenen Endanwenderzuteilung Rechnung, andererseits bilden

¹Die Erweiterungen sind farblich hervorgehoben.

sie die Grundlage für die mobile Ausnahmebehandlungsstrategie. An dieser Stelle sei nochmals erwähnt, dass ausschließlich interaktive mobile Aktivitäten betrachtet werden. Um die kontext-bezogene mobile Endanwenderzuteilung durchzuführen, muss zuerst der mobile Kontext (d.h. die manuell festzulegenden Parameter) *modelliert* werden. Danach werden bei der *Erzeugung* von Ausführungsinstanzen mobiler Aktivitäten mithilfe des *Ranking*-Algorithmus diejenigen mobilen Endanwender ausgewählt, die auf Grundlage des mobilen Kontexts am besten geeignet sind. Ferner ist ein entscheidender Aspekt, dass der *Ranking*-Algorithmus ohne manuelle Entscheidungen von mobilen Endanwendern durchgeführt wird. Insbesondere wäre eine manuelle Zuteilung zu komplex, da der mobile Kontext eine Vielzahl von Parametern umfasst. Zur *Ausführungszeit* wird der mobile Kontext schließlich verwendet, um die Ausführungsphase zu optimieren.

8.2 Modellierung mobiler Aktivitäten

Bei der Modellierung mobiler Aktivitäten besteht die Aufgabe für den Modellierer darin, die manuell zu bestimmenden Parameter des mobilen Kontexts festzulegen. Tabelle 8.1 zeigt diese Parameter im Überblick.

PM	Beschreibung	Art	PA	WB	obligat	MP
Parameterkategorie I						
SMG_{FF}	Formfaktor	Symbolwert	X	\checkmark	X	X
Parameterkategorie II						
MA_{GAO}	Geometrischer Ausführungsort	Messwert	$MA_{GAO} \rightarrow MA_{EB}$	\checkmark	X	\checkmark
MA_{SAO}	Symbolische Ausführungsorte	Symbolwert	X	\checkmark	X	\checkmark
MA_{EB}	Einzugsbereich	Symbolwert (Konstante)	X	\checkmark	X	X
MA_D	Dringlichkeit	Symbolwert (Konstante)	X	\checkmark	X	X
MA_{OFF}	Offline-Ausführungsmöglichkeit	Symbolwert (Konstante)	$MA_{OFF} \rightarrow MA_D$	\checkmark	X	X
MA_{GFF}	Geeigneter Formfaktor	Symbolwert	X	\checkmark	X	X
MA_{SWME}	Schwellwert Anzahl notwendiger mobiler Endanwender	Symbolwert (Konstante)	X	\checkmark	X	\checkmark
Parameterkategorie III						
GP_{ISZ}	Instant-Shutdown Referenzwert	Symbolwert	X	–	\checkmark	X
Parameterkategorie IV						
ME_{SAO}	Symbolische Aufenthaltsorte	Symbolwert	X	\checkmark	X	\checkmark
PM =Parameter, PA =Parameterabhängigkeit, WB =Wertebereich MP =Modellierungsprüfung, \checkmark =trifft zu, X =trifft nicht zu, –=nicht relevant						

Tabelle 8.1: Parametermodellierung

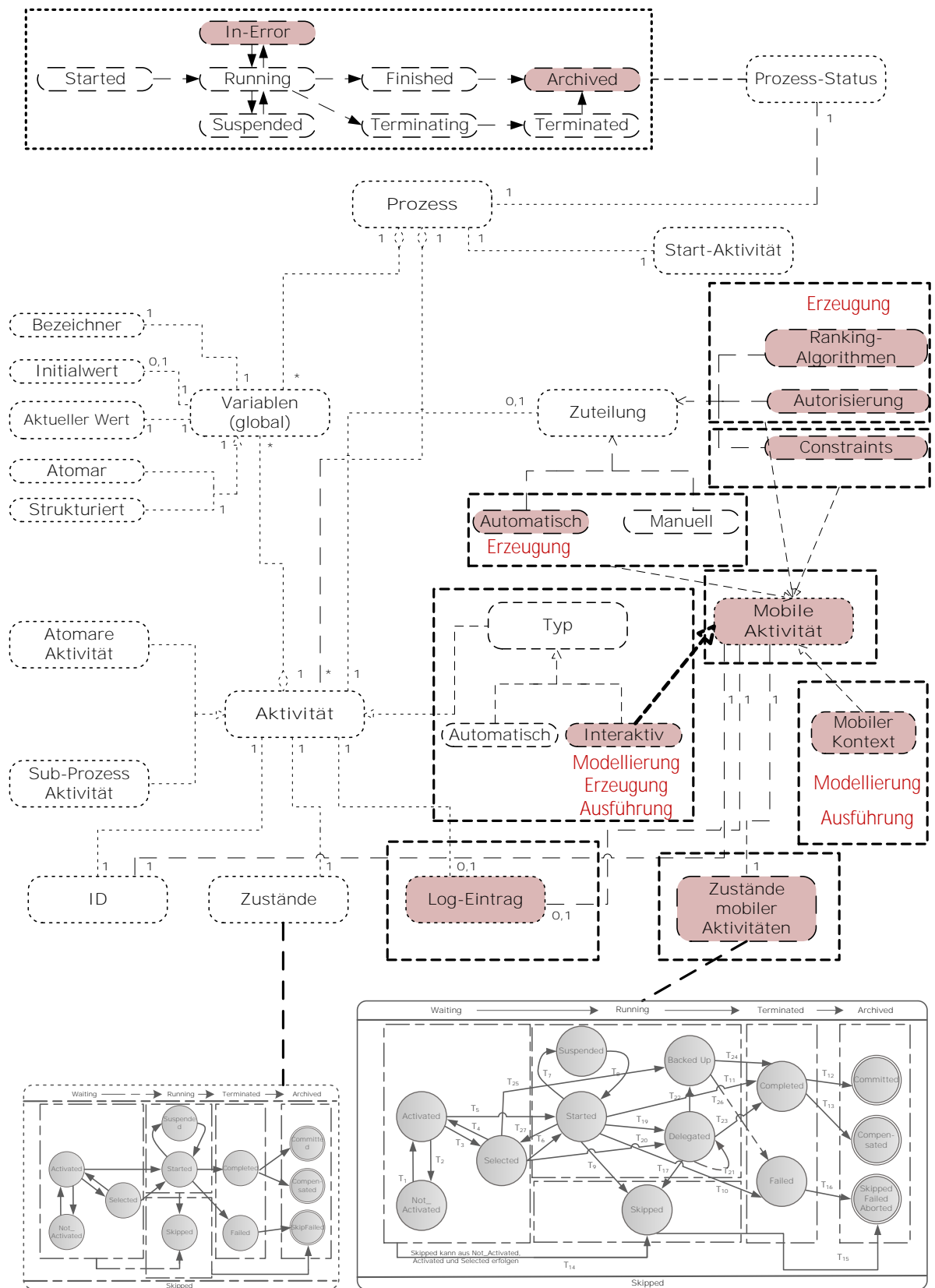


Abbildung 8.1: Erweiterung des Meta-Modells für mobile Aktivitäten

Tabelle 8.1 gibt weiters an, ob für einen Parameter ein vorgegebener Wertebereich (Spalte *WB*) existiert. Für den Parameter MA_{SAO} etwa könnte der Wertebereich $\{OP, Kreissaal, Schockraum\}$ lauten. Darüber hinaus gibt Tabelle 8.1 an, ob ein Parameter gesetzt (d.h. mit einem Wert versehen) werden muss (Spalte *obligat*). Wie Tabelle 8.1 zeigt, muss keiner der manuell festzulegenden Parameter zwingend mit einem Wert versehen werden. Zuletzt muss für den Parameter MA_{OFF} , sofern dieser gesetzt wird, die funktionale Abhängigkeit $MA_{OFF} \rightarrow MA_D$ spezifiziert werden. Diese gibt an, wie lange eine mobile Aktivität Offline ausgeführt werden darf (z.B. 5 Stunden), bevor die Ausnahmebehandlungsstrategie initiiert wird.

Ist die Festlegung der Parameter durch einen Modellierer abgeschlossen, werden ausgewählte Parameter mithilfe eines dazu entwickelten Algorithmus bewertet. Die Bewertung wiederum gibt einem Modellierer Rückmeldung, ob durch eine Veränderung der festgelegten Parameterwerte, eine weitere Verbesserung für die Zuteilung mobiler Endanwender zu mobilen Aktivitäten für die Ausführungszeit erreicht werden kann. Ob ein Parameter in die Bewertung des Algorithmus einfließt, zeigt die letzte Spalte in Tabelle 8.1.

Algorithmus 8.1 : Bestimmung von $OT(n)$

Data : Prozessmodell $PM = (N, E, \dots)$

Für alle Aktivitäten $n \in N$ die Menge potentiell qualifizierter Endanwender $PEA(n)$

Für alle Aktivitäten $n \in N$ die Schwellwerte notwendiger mobiler Endanwender $MA_{SWME}(n)$

Für alle Aktivitäten $n \in N$ die symbolischen Ausführungsorte $MA_{SAO}(n)$

Für alle Aktivitäten $n \in N$ die Datenabhängigkeit $GP_{DA}(n)$

Für alle Endanwender $ea \in PEA(n)$ einer mobilen Aktivität $n \in N$ die symbolischen Aufenthaltsorte $ME_{SAO}(ea)$

Result : $OT(n) = (GMEA, PEA, DA, SW)$: Tupel $OT(n)$ für alle mobile Aktivitäten $n \in N$

```

1  begin
2      /* Überprüfung aller mobilen Aktivitäten des betrachteten Prozesses */
3      foreach  $n \in N$  do
4          if  $n$  ist eine interaktive mobile Aktivität then
5               $OT(n) \leftarrow (0, 0, false, false);$  /*  $OT(n)$  initialisieren */
6               $SWZ \leftarrow 0;$  /* Zähler für geeignete mobile Endanwender initialisieren */
7              /* Menge geeigneter mobiler Endanwender durch symbolische Ortsübereinstimmung
8                  bestimmen */
9              foreach  $ea \in PEA(n)$  do
10                  if  $ME_{SAO}(ea) = MA_{SAO}(n)$  then
11                       $SWZ ++;$  /* Weiteren geeigneten mobilen Endanwender gefunden */
12                  end
13              end
14              /* Schwellwert  $MA_{SWME}(n)$  auf Erreichung überprüfen */
15              if  $SWZ \geq MA_{SWME}(n)$  then
16                  /* Schwellwert wurde erreicht */
17                   $OT(n) \leftarrow (SWZ, COUNT(PEA(n)), GP_{DA}(n), true)$ 
18              else
19                  /* Schwellwert wurde nicht erreicht */
20                   $OT(n) \leftarrow (SWZ, COUNT(PEA(n)), GP_{DA}(n), false)$ 
21              end
22          end
23      end
24  end

```

Algorithmus 8.1 berechnet für jede mobile Aktivität eines Prozesses ein Tupel $OT(n)$ bestehend aus vier Komponenten. Die erste Komponente gibt an, wie viele mobile Endanwender aus

dem Repository gefunden werden konnten, deren symbolischer Aufenthaltsort mit dem symbolischen Ausführungsort der mobilen Aktivität übereinstimmt. Die zweite Komponente gibt an, wie viele Endanwender generell für die mobile Aktivität infrage kommen, während die dritte Komponente ausdrückt, ob für die mobile Aktivität Datenabhängigkeiten zu nachfolgenden Aktivitäten bestehen. Die vierte Komponente zeigt schließlich an, ob der errechnete Wert der ersten Komponente größer als der Parameterwert MA_{SWME} ² ist. Ist letztgenannter Fall erfüllt, ist die vierte Komponente *true*, andernfalls ist sie *false*.

An dieser Stelle sei angemerkt, dass die Prüfung auf mögliche Datenabhängigkeiten zu nachfolgenden Aktivitäten im Algorithmus nicht im Detail gezeigt wird (vgl. Zeilen 11 und 13). Dies kann in [Rei00] nachgelesen werden.

Ein Beispieletupel $OT(n)$ für eine fiktive mobile Aktivität n könnte sich folgendermaßen darstellen: $OT(n) = (5, 15, true, false)$. D.h. für Aktivität n konnten 5 geeignete mobile Endanwender bezüglich des symbolischen Ausführungsortes gefunden werden konnten (Komponente 1). Ferner stehen insgesamt 15 potentielle Endanwender zur Verfügung (Komponente 2). Weiters existieren Datenabhängigkeiten zu nachfolgenden Aktivitäten (Komponente 3) und der Schwellwert für die notwendige Anzahl mobiler Endanwender konnte mit den ermittelten 5 Endanwendern nicht erreicht werden (Komponente 4).

Das Tupel $OT(n)$ bietet einem Modellierer folgende Bewertungsmöglichkeiten. Erstens erhält ein Modellierer durch Komponente 1 von $OT(n)$ Rückmeldung, ob die festgelegten Werte für die Parameter ME_{SAO} und MA_{SAO} für Aktivität n geschickt ausgewählt wurden. Zu diesem Zweck stellt Algorithmus 8.1 den Vergleich $ME_{SAO}(n) = MA_{SAO}(n)$ für alle mobilen Endanwender aus dem Repository an, die sich grundsätzlich zur Ausführungszeit einer Aktivität qualifizieren. Jede Übereinstimmung erhöht den Wert für Komponente 1 um 1. D.h. jede Erhöhung bedeutet, dass ein mobiler Endanwender aus dem Repository gefunden wurde, dessen symbolischer Aufenthaltsort mit dem symbolischen Ausführungsort der mobilen Aktivität n übereinstimmt.

Der Vergleich gibt daher Rückmeldung, ob sich eine große oder kleine Anzahl mobiler Endanwender nahe der mobilen Aktivität aufhalten. Ob die festgelegten Werte für die Parameter ME_{SAO} und MA_{SAO} für Aktivität n sinnvoll festgelegt wurden, kann ein Modellierer durch den Vergleich des Werts der Komponenten 1 und 2 entscheiden. Komponente 2 gibt die Anzahl aller mobilen Endanwender an, die sich zur Ausführungszeit für Aktivität n grundsätzlich qualifizieren. Ist der Wert von Komponente 2 größer als der von Komponente 1, können die festgelegten Werte für ME_{SAO} und MA_{SAO} prinzipiell verbessert werden.

Zweitens ersieht ein Modellierer anhand Komponente 4, ob der Schwellwert notwendiger mobiler Endanwender, die zur Ausführung der mobilen Aktivität n durch den Wert des Parameters MA_{SWME} gefordert werden, bereits durch den Wert von Komponente 1 erreicht wird. Ist der Wert von Komponente 1 größer als der von Komponente 4, besagt dies, dass der Schwellwert MA_{SWME} nicht nur erreicht wird, sondern alle ermittelten mobilen Endanwender zur Erreichung des Schwellwerts nahe der mobilen Aktivität n sind.

Drittens kann der Wert von Komponente 1 vor dem Hintergrund von Datenabhängigkeiten bewertet werden. Weist Komponente 1 einen niedrigen Wert für eine mobile Aktivität n mit Datenabhängigkeiten auf, sind nur wenige mobile Endanwender nahe dieser Aktivität. Speziell bei mobilen Aktivitäten mit Datenabhängigkeiten ist eine rasche Bearbeitung anzustreben. Eine rasche Bearbeitung ist für diejenigen mobilen Endanwender wahrscheinlicher, die sich nahe

²Zur Erinnerung: MA_{SWME} definiert den Schwellwert der mindestens geforderten mobilen Endanwender, die sich für eine mobile Aktivität zur Ausführungszeit qualifizieren.

der mobilen Aktivität aufhalten. So sollte für mobile Aktivitäten mit Datenabhängigkeiten die Anzahl mobiler Endanwender möglichst groß sein, die sich nahe der mobilen Aktivität aufhalten. Daher sollte der Wert von Komponente 1 in diesen Fällen möglichst groß sein. Abbildung 8.2 illustriert Beispielergebnisse für $OT(n)$.

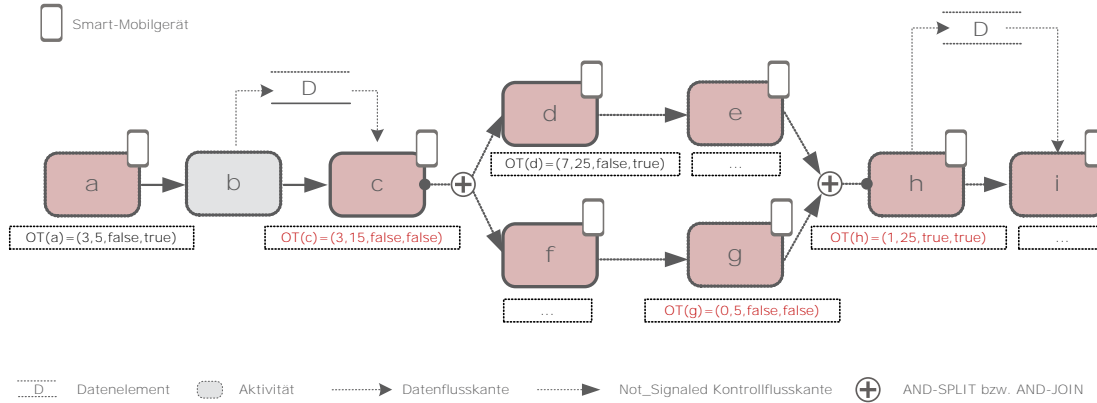


Abbildung 8.2: Beispielergebnisse für $OT(n)$

Die rot markierten Tupel $OT(n)$ aus Abbildung 8.2 illustrieren, dass die Festlegung der Werte für die Parameter ME_{SAO} und MA_{SAO} der entsprechenden mobilen Aktivitäten verbessert werden kann. Auf Basis dieser Angaben kann nun ein Modellierer entscheiden, ob er die festgelegten Werte für ME_{SAO} und MA_{SAO} ändern möchte. Nach jeder Änderung berechnet Algorithmus 8.1 erneut die Tupel $OT(n)$ aller mobilen Aktivitäten eines Prozesses. Dadurch erhält ein Modellierer Rückmeldung, ob die Änderung eine positive oder negative Änderung zur Folge hatte.

8.3 Erzeugung von Ausführungsinstanzen mobiler Aktivitäten

Bei der Erzeugung von Ausführungsinstanzen mobiler Aktivitäten wird die kontextbezogene mobile Endanwenderzuteilung angewandt. Der Grundbaustein dazu ist die Art der Zuteilung interaktiver Aktivitäten zu Endanwendern. Dies wurde für *PrMS* in Kapitel 6 vorgestellt. Das hier skizzierte Konzept wird nun für mobile Aktivitäten erweitert. Dazu werden bei der Erzeugung von Ausführungsinstanzen mobiler Aktivitäten nur noch diejenigen mobilen Endanwender berücksichtigt, die auf Grundlage des in der Folge vorgestellten *Ranking*-Algorithmus ausgewählt werden.

8.3.1 Mobiler Kontext

Der *Ranking*-Algorithmus wertet Parameter des mobilen Kontexts aus. Zwei Aspekte sind dabei besonders wichtig:

1. Zuerst muss festgelegt werden, welche Parameter des mobilen Kontexts vom *Ranking*-Algorithmus ausgewertet werden sollen. In unserem Ansatz sind dies alle Parameter des mobilen Kontexts mit Ausnahme von MA_{SFOS} und MA_{AZOS} (vgl. Tabelle 7.1).

2. Es wurde in Abschnitt 7.2 für den mobilen Kontext gezeigt, dass auf Basis der verschiedenen Parameter insgesamt 4 Regeltypen existieren. Um den *Ranking*-Algorithmus anzuwenden, sind exakt diese 4 Regeltypen von Interesse (vgl. Tabelle 8.2).

	Ranking	PrMS
Parametervergleiche		
$0 < normEZB(ME) \leq 1$	✓	X
$MA_{SAO} = ME_{SAO}$	✓	X
$MA_{GFF} = SMG_{FF}$	✓	X
$SMG_{BLS} \geq MA_{NBLS}$	✓	X
$ME_{ISZ} \geq GP_{ISZ}$	✓	X
Präzedenzen		
(1) $0 < normEZB(ME) \leq 1$	–	–
(1) $SMG_{BLS} \geq MA_{NBLS}$	–	–
(2) $MA_{SAO} = ME_{SAO}$	–	–
(4) $MA_{GFF} = SMG_{FF}$	–	–
(5) $ME_{ISZ} \geq GP_{ISZ}$	–	–
Formeln		
$normEZB(ME) = \begin{cases} \frac{MA_{GAO} - SMG_{GAO}(ME)}{MA_{EB}}, & \text{wenn geometrische Daten bestimmbar} \\ 0, & \text{wenn geometrische Daten nicht bestimmbar} \end{cases} \quad (8.1)$		
Funktionale Abhängigkeiten		
$SMG_{ONT} \rightarrow SMG_{BLS}$	X	✓
$MA_{GAO} \rightarrow MA_{EB}$	X	✓
<i>X=trifft nicht zu, ✓=trifft zu, –=nicht relevant</i>		

Tabelle 8.2: Kontextregeln für Ranking-Algorithmus

Tabelle 8.2 gibt zudem an, ob das Ergebnis einer Regelauswertung vom *Ranking*-Algorithmus berechnet (d.h. für die Spalte *Ranking* = ✓ gilt) oder vom *PrMS* zur Verfügung gestellt wird (d.h. für die Spalte *PrMS* = ✓ gilt). Zuletzt sei nochmals auf die Funktionsweise funktionaler Abhängigkeiten zwischen Parametern hingewiesen (vgl. Tabelle 8.2). Diese dienen dazu, aus dem Wert eines Parameters automatisch den Wert eines zweiten Parameters abzuleiten. Im Beispiel $SMG_{ONT} \rightarrow SMG_{BLS}$ aus Tabelle 8.2 drückt die funktionale Abhängigkeit aus, dass der angemeldete Netztyp eines Smart-Mobilgeräts SMG_{ONT} dessen geforderten Batterieladestatus SMG_{BLS} des Smart-Mobilgeräts bestimmt. Dadurch bestimmt der Parameter SMG_{ONT} funktional wie hoch der Batterieladestatus des Smart-Mobilgeräts sein muss, damit dessen mobiler Endanwender des Smart-Mobilgeräts für die Ausführung einer mobilen Aktivität berücksichtigt wird. Weiter sei darauf hingewiesen, dass die funktionalen Abhängigkeiten einmalig und auf Basis von Erfahrungswerten im Repository des *PrMS* hinterlegt werden. Funktionalen Abhängigkeiten müssen nicht manuell festgelegt werden, sie stellen daher einen wichtigen Aspekt der kontextbezogenen mobilen Endanwenderzuteilung dar.

8.3.2 Ranking-Algorithmus

Mithilfe der erfassten Parameter (vgl. Tabelle 7.1) und der Regeltypen (vgl. Tabelle 8.2) sind alle relevanten Informationen für den *Ranking*-Algorithmus zusammengetragen. Im Folgenden

wird der Algorithmus im Detail vorgestellt.

Konkret bestimmt dieser Algorithmus eine Liste, die mit $KMEZL(n)$ bezeichnet wird. Nur in dieser Liste enthaltene mobile Endanwender werden bei der Erzeugung einer Ausführungsinstanz einer mobilen Aktivität n berücksichtigt. Das primäre Ziel des entwickelten Algorithmus ist es, den mobilen Kontext bestmöglich zu nutzen, sodass nur diejenigen mobilen Endanwender ausgewählt (d.h. geranked) werden, die der mobilen Aktivität möglichst nahe sind und die zusätzlich die besten Parameterwerte bezüglich der Parameter SMG_{BLS} , SMG_{FF} , GP_{DA} und MA_D inne haben.

Algorithmus 8.2 : Primäre Bestimmung von $KMEZL(n)$

Data : Relevante Parameterwerte des mobilen Kontexts (vgl. Tabelle 7.1)

Regeln des mobilen Kontexts (vgl. Tabelle 8.2)

$MAME \subseteq PEA(n)$: Menge angemeldeter und berechtigter mobiler Endanwender

Result : $KMEZL(n)$: Menge geeigneter mobiler Endanwender zur Erzeugung einer Ausführungsinstanz der mobilen Aktivität n

```

1  begin
2       $KMEZL(n) \leftarrow \emptyset$ ;                /* Menge geeigneter mobiler Endanwender initialisieren */
3       $GMEAZ \leftarrow 0$ ;                    /* Zähler geeigneter mobiler Endanwender initialisieren */
4       $GFFZ \leftarrow 0$ ;                    /* Zähler für Parametervergleich auf den Formfaktor initialisieren */
      /* Matching auf geometrische Ortsübereinstimmung und Batterieladestatus */
5      foreach Endanwender  $ea \in MAME$  do
6          if  $(0 < normEZB(ME(ea)) \leq 1) \wedge (SMG_{BLS}(ea) \geq MA_{NBLS}(n))$  then
7               $GMEAZ++$ ;
8               $KMEZL(n) \leftarrow KMEZL(n) \cup \{ea\}$ ;
9              if  $MA_{GFF}(n) = SMG_{FF}(ea)$  then
10                  $GFFZ++$ ;
11             end
12         end
13     end
      /* Matching auf symbolische Ortsübereinstimmung und Batterieladestatus */
14     if  $GMEAZ < MA_{SWME}(n)$  then
15         foreach Endanwender  $ea \in (MAME - KMEZL(n))$  do
16             if  $(MA_{SAO}(n) = ME_{SAO}(ea)) \wedge (SMG_{BLS}(ea) \geq MA_{NBLS}(n))$  then
17                  $GMEAZ++$ ;
18                  $KMEZL(n) \leftarrow KMEZL(n) \cup \{ea\}$ ;
19                 if  $MA_{GFF}(n) = SMG_{FF}(ea)$  then
20                      $GFFZ++$ ;
21                 end
22             end
23         end
24     end
      /* Spezielle Berücksichtigung der Datenabhängigkeit und der Dringlichkeit */
25     if  $(GP_{DA}(n) \vee MA_D \neq 0) \wedge GFFZ > MA_{SWME}(n)$  then
26         foreach Endanwender  $ea \in KMEZL(n)$  do
27             if  $MA_{GFF}(n) \neq SMG_{FF}(ea)$  then
28                  $KMEZL(n) \leftarrow KMEZL(n) \cap \{ea\}$ ;
29             end
30         end
31     end
32 end
    
```

Es kann vorkommen, dass Algorithmus 8.2 eine leere Liste $KMEZL(n)$ zurückgibt. Da der Algorithmus primär auf eine passende Ortsübereinstimmung abzielt (s. Schleifen in Zeilen 5

und 15) und zusätzlich den höchsten Batterieladestatus berücksichtigt (s. Vergleiche in Zeilen 6 und 16), kann diese Berechnung ergebnislos sein. In einem solchen Fall wird die Liste $KMEZL(n)$ durch einen abgewandelten *Ranking*-Algorithmus (s. Algorithmus 8.3) berechnet. Die Überlegung zu Algorithmus 8.3 besteht darin, nur diejenigen mobilen Endanwender für die Zuteilung der mobilen Aktivität auszuwählen, deren Batterieladestatus hoch und deren Instant-Shutdown-Zähler niedrig ist (s. Zeile 7 Algorithmus 8.3). Weiters wird der Parameter MA_{SWME} herangezogen (vgl. Zeile 21 Algorithmus 8.3), um zu bestimmen, welche mobilen Endanwender bezogen auf den Batterieladestatus und Instant-Shutdown-Zähler besser geeignet sind als andere. Da Algorithmus 8.3 im Vergleich zu Algorithmus 8.2 den Ausführungsort überhaupt nicht berücksichtigt, findet er generell mehr mobile Endanwender und wird daher zur sekundären Bestimmung angewandt. Nichtsdestotrotz berücksichtigt auch Algorithmus 8.3 bezüglich der genannten Parameter die kontextbezogene Zuteilung mobiler Endanwender zu mobilen Aktivitäten. Daher kann dieser als Variante zu Algorithmus 8.2 angewandt werden.

Wie Algorithmus 8.2 zur primären Berechnung von $KMEZL(n)$ arbeitet, soll noch grafisch illustriert werden. Dazu zeigt Abbildung 8.3 zwei Beispielauswertungen (vgl. Szenarien I und II). In Szenario I errechnet Algorithmus 8.2, dass sich der mobile Endanwender *Schickler* zur Ausführung der mobilen Aktivität $m1$ eignet ①, während er sich für die mobile Aktivität $m2$ nicht ② eignet. In Szenario II wiederum errechnet Algorithmus 8.2, dass sich der mobile Endanwender *Tiedeken* nicht ③ zur Ausführung der mobilen Aktivität $m5$ eignet.

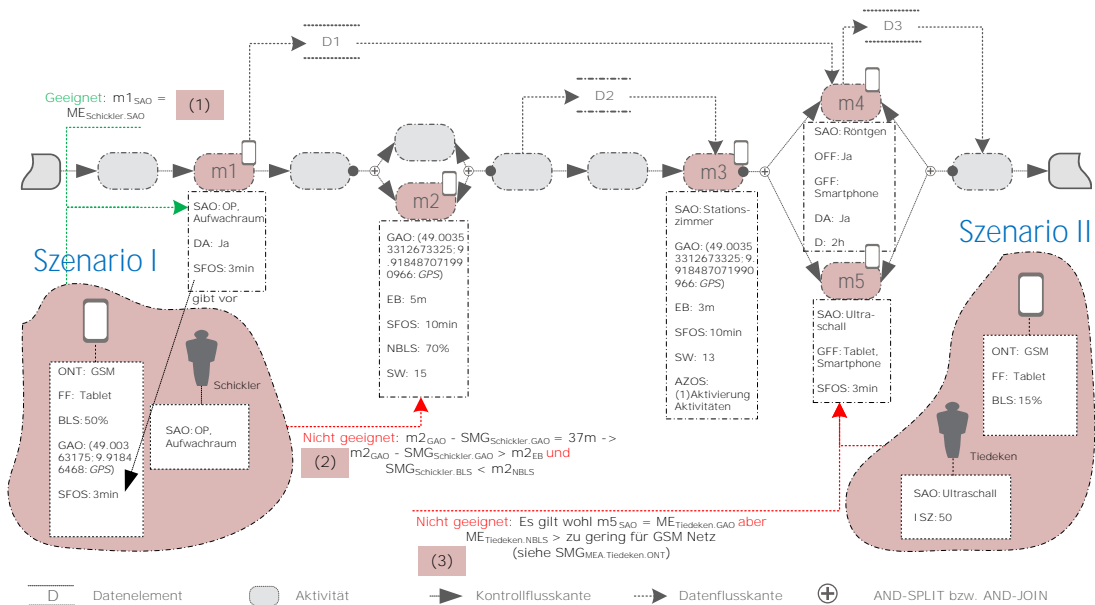


Abbildung 8.3: Beispielauswertungen des primären *Ranking*-Algorithmus

8.3.3 Mobiler Kontext und Worklist-Sortierung

Nachdem ein mobiler Endanwender eine mobile Aktivität angenommen hat, wird diese in seine Liste angenommener Aktivitäten übernommen. Bis er die Aktivität auch tatsächlich ausführt, kann weitere Zeit verstreichen. Um dem mobilen Endanwender anzuzeigen, welche Aktivitäten rascher bearbeitet werden sollten, wird abermals der mobile Kontext herangezogen. Dazu

Algorithmus 8.3 : Sekundäre Bestimmung von $KMEZL(n)$

Data : Relevante Parameterwerte des mobilen Kontexts (vgl. Tabelle 7.1)

Regeln des mobilen Kontexts (vgl. Tabelle 8.2)

 $MAME \subseteq PEA(n)$: Menge angemeldeter und berechtigter mobiler Endanwender

Result : $KMEZL(n)$: Menge geeigneter mobiler Endanwender zur Erzeugung einer Ausführungsinstanz der mobilen Aktivität n

```

1  begin
2       $KMEZL(n) \leftarrow \emptyset$ ;                                /* Menge geeigneter mobiler Endanwender initialisieren */
3      if  $|MAME| > MASWME(n)$  then
4           $KMEZLH \leftarrow \emptyset$ ;                            /* Hilfsliste */
5           $EW \leftarrow \emptyset$ ;                                /* Liste für Eignungswert mobiler Endanwender initialisieren */
6           $cai \leftarrow \emptyset$ 
7          foreach Endanwender  $ea \in MAME$  do
8               $EW[cai] \leftarrow SMGBLS(ea) + \frac{1}{MEISZ(ea)}$ ;
9               $KMEZLH \leftarrow KMEZLH \cup \{ea\}$ ;
10              $cai++$ ;
11         end
12         /*  $KMEZLH, KMEZLH[1], \dots, KMEZLH[m]$  gemäß  $EW$  aufsteigend sortieren */
13          $tmp \leftarrow \emptyset$ ;                                /* TMP-Variable für einzusortierenden Wert initialisieren */
14          $tmp2 \leftarrow \emptyset$ ;                                /* TMP-Variable für einzusortierenden Wert initialisieren */
15          $j \leftarrow 0$ ;                                        /* Hilfszähler initialisieren */
16         for  $i \leftarrow 1$  to  $|KMEZLH|$  do
17              $tmp \leftarrow KMEZLH[i]$ ;
18              $tmp2 \leftarrow EW[i]$ ;
19              $j \leftarrow i$ ;
20             while  $(j > 1 \wedge KMEZLH[j-1] < EW[i])$  do
21                  $KMEZLH[j] \leftarrow KMEZLH[j-1]$ ;
22                  $EW[j] \leftarrow EW[j-1]$ ;
23                  $j \leftarrow j-1$ ;
24                  $KMEZLH[j] \leftarrow tmp$ ;
25                  $EW[j] \leftarrow tmp2$ ;
26             end
27         end
28         /* Obersten  $MASWME(n)$  Einträge aus  $KMEZLH$  in  $KMEZL(n)$  überführen */
29         for  $i \leftarrow 1$  to  $MASWME(n)$  do
30              $KMEZL(n) \leftarrow KMEZL(n) \cup KMEZLH[i]$ ;
31         end
32     else
33         foreach Endanwender  $ea \in MAME$  do
34              $KMEZL(n) \leftarrow KMEZL(n) \cup \{ea\}$ ;
35         end
36     end
37 end

```

werden die angenommenen mobilen Aktivitäten in der Worklist des mobilen Endanwenders priorisiert. Aktivitäten mit einer höheren Priorität als andere zeigen dem mobilen Endanwender an, dass diese vor dem Hintergrund einer robusten Ausführung rascher bearbeitet werden sollten. Beispielsweise sollten mobile Aktivitäten, die Daten schreiben, möglichst rasch und zuverlässig ausgeführt werden.

Die Priorisierung erfolgt in zwei Phasen. Die erste Phase findet direkt nach der Annahme einer mobilen Aktivität durch einen mobilen Endanwender statt. Die zweite Phase überprüft peri-

odisch die Liste aller angenommenen mobilen Aktivitäten eines mobilen Endanwenders. Für beide Phasen kommen die folgenden Regeln zum Einsatz:³

1. *Regel 1*: Besteht für die angenommene mobile Aktivität eine Datenabhängigkeit (d.h. $GP_{DA}(n) = true$), wird die Priorität um 1 erhöht.
2. *Regel 2*: Besteht für die angenommene mobile Aktivität eine Dringlichkeit (d.h. $MA_D(n) > 0$), wird die Priorität um 1 erhöht.
3. *Regel 3**: Ist für die angenommene mobile Aktivität ein notwendiger Batterieladestatus gesetzt (d.h. $MA_{NBLS}(n) > 0$) und gilt für das Smart-Mobilgerät des mobilen Endanwenders $SMG_{BLS} < MA_{NBLS}$, wird die Priorität um 1 erhöht. Die Priorität wird nur nach dem ersten Überschreiten erhöht und wieder reduziert, wenn $SMG_{BLS} > MA_{NBLS}$ gilt.
4. *Regel 4**: Wurde für eine mobile Aktivität ein geometrischer Ausführungsort ermittelt und gilt dies auch für den mobilen Endanwender (d.h. $normEZZB(ea) > 0$), wird die Priorität um 1 erhöht. Ferner erhält in diesem speziellen Fall die Aktivität eine Markierung, die dem mobilen Endanwender anzeigt, dass er sich nahe der mobilen Aktivität aufhält. Die Maßnahme der Markierung hat sich aus Praxistests ergeben.

Um die Priorisierung für die Einträge angenommener mobiler Aktivitäten festzulegen, werden die im Kapitel 6 vorgestellten Operationen angewandt. Generell dient die Priorisierung dem Zweck, mobilen Endanwendern einen schnellen Überblick zu geben, welche Aktivitäten rascher ausgeführt werden sollten als andere. Der mobile Endanwender muss dazu keinen Einblick in die Parametrisierung haben. Nichtsdestotrotz wird ihm überlassen, in welcher Reihenfolge er seine mobilen Aktivitäten ausführt. Die Priorisierung ist ein weiterer Baustein der kontextbezogenen mobilen Endanwenderzuteilung.

Abschließend sei erwähnt, dass die Priorisierung auch für noch nicht zugeteilte mobile Aktivitäten angewandt wird. Damit erhält ein mobiler Endanwender auch für diese Aktivitäten einen raschen Überblick über die Parameter des mobilen Kontexts.

8.4 Ausführung mobiler Aktivitäten

Auch während der Ausführung einer mobilen Aktivität wird der mobile Kontext genutzt. Die Ausführung beginnt, sobald ein mobiler Endanwender die mobile Aktivität startet; d.h. sie aus der Liste angenommener Aktivitäten zur Ausführung auswählt (vgl. Kapitel 6). Welche Parameter und Regeln des mobilen Kontexts während der Ausführung exakt verwendet werden, zeigt Tabelle 8.3.

Bezogen auf Tabelle 8.3 erfordern zwei Aspekte eine gesonderte Diskussion:

1. Die beiden Parameter MA_{SFOS} und MA_{AZOS} dienen dazu, dass möglichst rasch erkannt werden kann, ob ein Smart-Mobilgerät nicht mehr Online ist. Dieser Aspekt ist für die Ausnahmebehandlung entscheidend. Jedoch ist die Festlegung dieser Parameter nur auf Grundlage von Erfahrungswerten möglich. Weiters haben Studien gezeigt [PHZ12], dass die Überprüfung des Online-Status viele I/O-Zugriffe auf einem Smart-Mobilgerät

³Ist die Regel mit einem Stern gekennzeichnet, wird sie periodisch ausgeführt, andernfalls einmal nach Annahme der mobilen Aktivität.

Erläuterungen	
Parameter	
ME_{ISZ}	Wird erhöht, wenn der mobile Endanwender einen Instant-Shutdown durchführt.
MA_D	Bei gesetzter Dringlichkeit darf die mobile Aktivität nicht suspendiert (d.h. <i>suspended</i>) werden (vgl. Kapitel 5).
MA_{OFF}	Bei gesetzter Offline-Funktionalität werden MA_{SFOS} und MA_{AZOS} auf 0 gesetzt.
GP_{DA}	Wird ausgewertet wegen der funktionalen Abhängigkeit $GP_{DA} \rightarrow MA_{SFOS}$.
MA_{SFOS}	Zur Statusüberprüfung des Online-Status des Smart-Mobilgeräts.
MA_{AZOS}	Zur Statusüberprüfung des Online-Status des Smart-Mobilgeräts.
SMG_{ONT}	Wird ausgewertet wegen funktionaler Abhängigkeit $SMG_{ONT} \rightarrow SMG_{BLS}$.
MA_{NBLS}	Wird ausgewertet wegen funktionaler Abhängigkeit $MA_{NBLS} \rightarrow MA_{SFOS}$.
Parametervergleiche	
$0 < normEZB(ME) \leq 1$	Dient der Priorisierung mobiler Aktivitäten (vgl. Abschnitt 7)
$SMG_{BLS} \geq MA_{NBLS}$	Dient der Priorisierung mobiler Aktivitäten (vgl. Abschnitt 7)
Formeln	
$normEZB(ME) = \begin{cases} \frac{MA_{GAO} - SMG_{GAO}(ME)}{MA_{EB}}, & \text{wenn geometrische Daten bestimmbar} \\ 0, & \text{wenn geometrische Daten nicht bestimmbar} \end{cases} \quad (8.2)$	
Funktionale Abhängigkeiten	
$SMG_{ONT} \rightarrow SMG_{BLS}$	—
$MA_{GAO} \rightarrow MA_{EB}$	—
$MA_{OFF} \rightarrow MA_D$	—
$MA_{OFF} \rightarrow MA_{SFOS}$	—
$MA_{NBLS} \rightarrow MA_{SFOS}$	—
$GP_{DA} \rightarrow MA_{SFOS}$	—
— = keine Erläuterungen notwendig	

Tabelle 8.3: Kontextparameter zur Ausführung mobiler Aktivitäten

erfordert. Die Zugriffe wiederum verursachen einen erhöhten Batterieverbrauch. Je häufiger daher der Online-Status überprüft wird, desto mehr Energie wird auf einem Smart-Mobilgerät verbraucht. Eine häufige Abfrage des Online-Status gewährleistet allerdings, dass ein Offline-Zustand rasch erkannt wird. Daher sollte abhängig vom konkreten Szenario ein guter Kompromiss zwischen Online-Statusüberprüfung und Batterieverbrauch gewählt werden.

- Die beiden Parametervergleiche aus Tabelle 8.3 werden bei der Ausführung mobiler Aktivitäten dazu verwendet, periodisch die in Abschnitt 8.3.3 gezeigte Priorisierung zu realisieren. Daher findet ein mobiler Endanwender auch nach Fertigstellung einer mobilen Aktivität stets eine priorisierte Worklist vor.

8.4.1 Systemwechsel

Eine weitere Anforderung besteht darin, dass ein Wechsel der Ausführung einer mobilen Aktivität vom Smart-Mobilgerät zu einem Desktop-System möglich sein sollte. Der realisierte Systemwechsel muss von einem mobilen Endanwender explizit an das *PrMS* als Anfrage gestellt werden. Diese Anfrage kann er während der Ausführungszeit stellen bzw. bevor er die mobile Aktivität startet, d.h. während die mobile Aktivität sich im Zustand *Selected* befindet. Möchte ein mobiler Endanwender einen Systemwechsel durchführen, wird vom *PrMS* eine weitere Aktivität im Prozessablauf eingefügt (vgl. Abb. 8.4), die den Systemwechsel realisiert. Entscheidet sich der mobile Endanwender für den Systemwechsel, werden die mobile Aktivität beendet und die neu hinzugefügte Aktivität aktiviert. Danach wartet das *PrMS* bis sich der mobile Endanwender auf einem Desktop-System angemeldet hat. Ist dies der Fall, wird die Aktivität in seine Worklist übertragen und dort in den Zustand *Selected* versetzt. Die Ein- und Ausgabedaten für die neue Aktivität ergeben sich aus den Ein- und Ausgabedaten der mobilen Aktivität. Wurden von der mobilen Aktivität bereits Daten erzeugt, werden diese Daten von der neuen Aktivität gelesen.

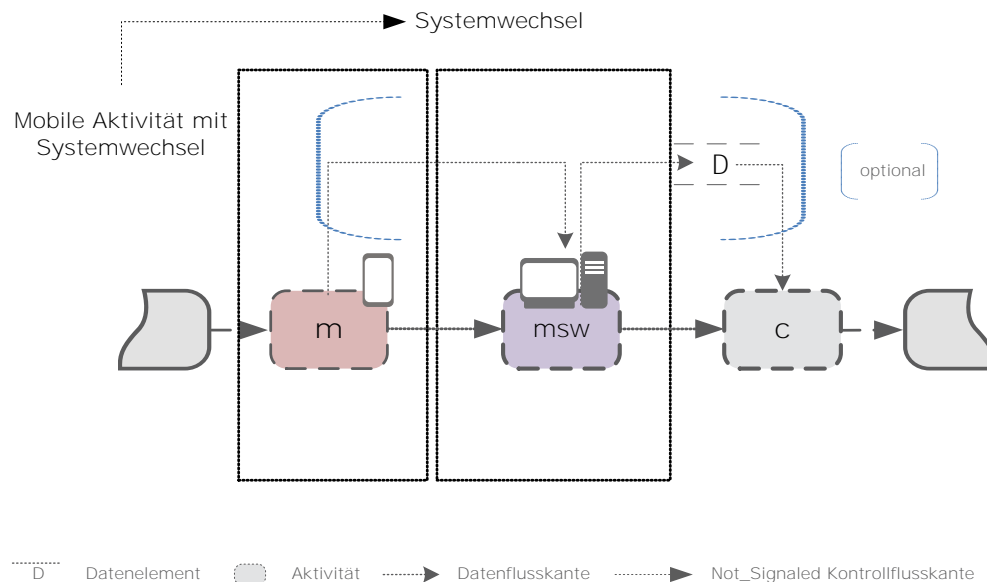


Abbildung 8.4: Systemwechsel

Zum Systemwechsel sind folgende Punkte gesondert zu erwähnen. *Erstens* erfolgt der Systemwechsel stets vom Smart-Mobilgerät zum Desktop-System. Der umgekehrte Weg wird in dieser Arbeit nicht berücksichtigt, da weitere Überlegungen für den mobilen Kontext notwendig wären, die den Rahmen der Arbeit übersteigen würden. *Zweitens* wird der Systemwechsel über eine neue Aktivität realisiert, da mit dieser Maßnahme geringe Eingriffe in ein *PrMS* notwendig werden. Selbst das Ausführungsprotokoll muss nicht geändert werden, da die mobile Aktivität ordnungsgemäß beendet wird und die neue Aktivität aus Sicht des *PrMS* ohne Änderungen ausgeführt werden kann. Obwohl dynamische Strukturänderungen zur Ausführungszeit mit technischem Aufwand verbunden sind, der wiederum erheblich sein kann [WRRM08], ist dieser Aufwand für den Systemwechsel vertretbar. Da nachfolgende Aktivitäten von dieser Änderung

unberührt bleiben, ist der erzeugte Aufwand gering. Ferner wird durch diese Realisierung des Systemwechsels gewährleistet, dass dieser explizit in der Ausführungshistorie vermerkt wird. *Drittens* beugt der Systemwechsel proaktiv einer Ausnahmebehandlung vor, die der robusten Ausführung mobiler Aktivitäten dienlich ist. Mit dem Zeitpunkt des Systemwechsels kann keine Ausnahme bei der Ausführung der mobilen Aktivität mehr erzeugt werden. Wird jedoch eine Ausnahme der mobilen Aktivität vor dem Systemwechsel erzeugt, dann wird weiterhin die Ausnahmebehandlungsstrategie angewandt. Im letztgenannten Fall kann ein Systemwechsel die robuste Ausführung daher nicht fördern. *Viertens* setzt der Systemwechsel voraus, dass der mobile Endanwender Rechte besitzt, sich auch an Desktop-Systemen anmelden zu können. In dieser Arbeit wird davon ausgegangen, dass sich diese stets an Desktop-Systemen anmelden können.

8.5 Weitere Aspekte

Nachfolgend werden weitere Aspekte im Kontext der Modellierung, Erzeugung und Ausführung mobiler Aktivitäten diskutiert.

Für die Berechnung des *Ranking*-Algorithmus wird vorausgesetzt, dass sich aus der Menge $PEA(n)$ (vgl. Kapitel 5) stets mobile Endanwender zur Erzeugung und Ausführung einer mobilen Aktivität ermitteln lassen. Was aber ist zu tun, wenn sich kein Endanwender aus $PEA(n)$ mobil angemeldet hat bzw. verfügbar ist. In diesem Fall gibt es zwei mögliche Ansätze für das weitere Vorgehen:

1. Aktivierung der Ausnahmebehandlungsstrategie.
2. Periodische Anwendung des *Ranking*-Algorithmus, bis sich wieder mobile Endanwender bestimmen lassen.

Der zweite Ansatz wird von der kontextbezogenen mobilen Endanwenderzuteilung verwendet und hat sich in der Praxis auch bewährt. Anzumerken ist ferner, dass sowohl der primäre als auch sekundäre *Ranking*-Algorithmus periodisch angewendet werden. Als weitere Maßnahme kann eine Delegation von Rechten erfolgen, um Endanwender, die gerade Online sind, aber keine Ausführungsrechte haben, temporär mit solchen Rechten auszustatten. In der Folge erhält man eine größere Anzahl qualifizierter mobiler Endanwender. Dieses Prinzip kann in [CK08b, CK08c, CK08a, WKB07, AW05, GSFC08, BVA06, KGG⁺06, SLS06, VO02, PRMD08] weiter vertieft werden. Es wird in dieser Arbeit nicht weiter verfolgt.

In Kapitel 5 wurde gezeigt, welche Strukturen das verwendete Prozess-Metamodell ermöglicht. Für drei dieser Strukturen sind weitere Betrachtungen im Zusammenhang mit der kontextbezogenen mobilen Endanwenderzuteilung notwendig. Es handelt sich um *Verzweigungen*, *Sync-Kanten* und *Schleifen* (vgl. Abb. 8.5). Befindet sich die mobile Aktivität an den mit Ausrufezeichen markierten Stellen (vgl. Abb. 8.5), wird die mobile Aktivitäten so behandelt als ob eine Datenabhängigkeit besteht. In der Folge werden von der kontextbezogenen mobilen Endanwenderzuteilung diejenigen Maßnahmen angewandt, die auch im Falle einer Datenabhängigkeit zum Tragen kommen. Dadurch wird sichergestellt, dass die nachfolgenden Aktivitäten rascher und damit robuster ausgeführt werden können, da durch die *künstliche* Datenabhängigkeit, die am besten geeigneten mobilen Endanwender zur Ausführung der mobilen Aktivität ausgewählt werden.

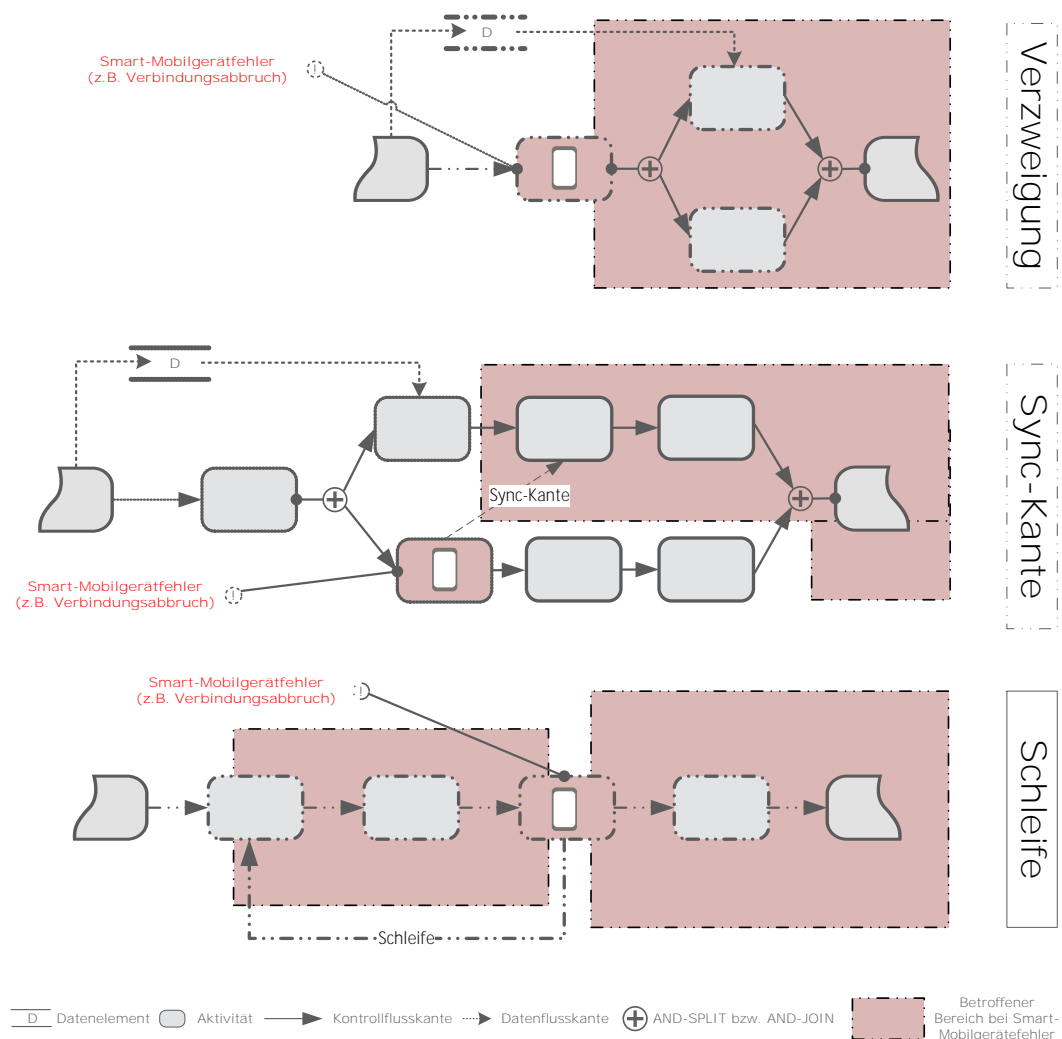


Abbildung 8.5: Berücksichtigung von Prozess-Strukturen

In Kapitel 6 wurde das Protokoll vorgestellt, welches zwischen *PrMS* und Prozess-Klient bei der Aktivitätsausführung verwendet wird. In diesem Abschnitt wird gezeigt, an welchen Stellen des Protokollablaufs die Parameter des mobilen Kontexts vom Smart-Mobilgerät an das *PrMS* übertragen werden (vgl. Abb. 8.6 ①-④). Tabelle 8.4 führt die entsprechenden Parameter bezogen auf die markierten Stellen auf. Auf Basis dieser Übertragungen hat das *PrMS* stets einen aktuellen Status des Smart-Mobilgeräts. Abschließend sei erwähnt, dass über die Parameter *MA_{SFOS}* und *MA_{AZOS}* festgelegt wird, wie oft Schritte 6 - 9 (vgl. Abb. 8.6) durchgeführt werden.

Zuletzt sollen Ad-hoc Änderungen diskutiert werden. Da Ad-hoc Änderungen die Flexibilität eines *PrMS* steigern (vgl. Kapitel 5), muss dieser Aspekt vor dem Hintergrund mobiler Aktivitäten ebenfalls diskutiert werden. Ad-hoc Änderungen sind mit den vorgestellten Konzepten kompatibel. Die einzige Einschränkung besteht darin, dass für mobile Aktivitäten, die ad-hoc hinzugefügt werden sollen, ihr mobiler Kontext bereits vormodelliert vorliegen soll. D.h. es muss

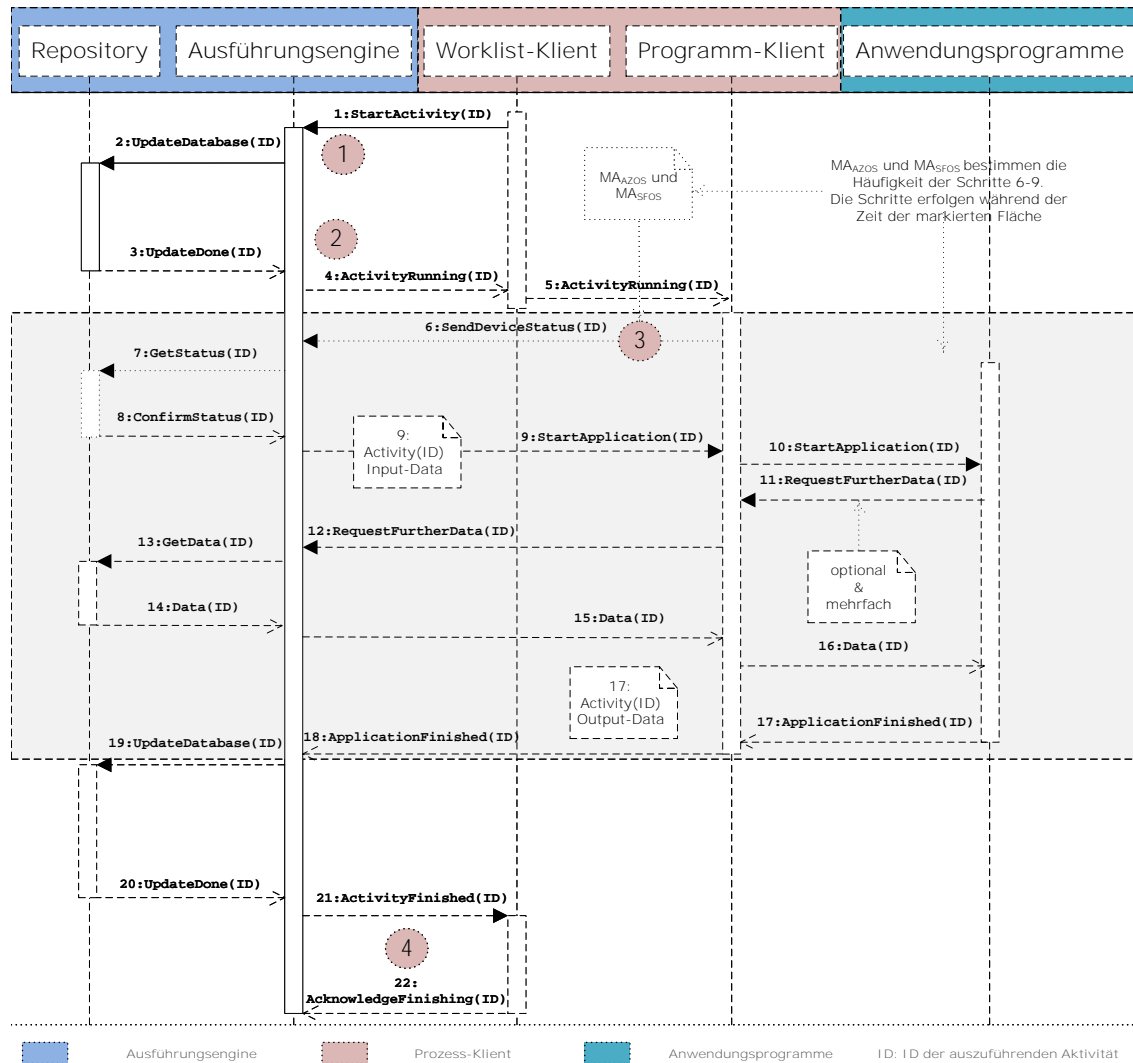


Abbildung 8.6: Ausführungsprotokoll und Parameterübertragung

Protokollstelle	Parameter
①	SMG_{BLS} , SMG_{ONT} , SMG_{GAO} , interne SMG -Parameter wegen Instant-Shut downzählung
②	MA_{SFOS} , MA_{AZOS}
③	SMG_{BLS} , SMG_{ONT} , SMG_{GAO} , interne SMG -Parameter wegen Instant-Shut downzählung
④	SMG_{BLS} , SMG_{ONT} , SMG_{GAO} , interne SMG -Parameter wegen Instant-Shut downzählung

Tabelle 8.4: Übertragene Parameter

im Repository des *PrMS* hinterlegt sein, welche Parameter für die mobile Aktivität, die spon-

tan hinzugefügt werden soll, gelten. Ist der letztgenannte Aspekt sichergestellt, können Ad-hoc Änderungen ohne weitere Maßnahmen angewandt werden.

8.6 Diskussion

Nachfolgend wird diskutiert, welcher Beitrag durch die kontextbezogene mobile Endanwenderzuteilung zur robusten Ausführung mobiler Aktivitäten erreicht wurde. Andererseits werden verwandte Ansätze diskutiert.

8.6.1 Beitrag zur robusten Ausführung mobiler Aktivitäten

Die vorgestellten Konzepte dieses Kapitels erfüllen mehrere Anforderungen (vgl. Kapitel 2), denen sich diese Arbeit stellt. Tabelle 8.5 fasst diese zusammen und zeigt, welche Maßnahmen zur Erfüllung der jeweiligen Anforderung beitragen. So zeigt Tabelle 8.5, dass 13 der gestellten Anforderungen durch die in diesem Kapitel vorgestellten Konzepte erfüllt werden. Insbesondere werden die funktionalen Anforderungen des mobilen Kontexts erfüllt (Anforderungen 10-15), sowie das frühzeitige Erkennen von Laufzeitproblemen (Anforderungen 18). Letztere erfüllt das proaktive Erkennen von Laufzeitproblemen (vgl. Abschnitt 7.1).

Anforderung	Maßnahmen
Funktionale Anforderungen mobiler Aktivitäten	
Anforderung 1	Die Konzepte berücksichtigen den mobilen Kontext.
Anforderung 3	Die Konzepte realisieren eine automatische Priorisierung mobiler Aktivitäten (vgl. Abschnitt 8.3.3).
Anforderung 4	Die Konzepte unterstützen das Unterbrechen mobiler Aktivitäten.
Anforderung 6	Die Konzepte unterstützen einen Systemwechsel (vgl. Abschnitt 8.4.1).
Anforderung 7	Die Konzepte unterstützen das spontane Hinzufügen und Löschen mobiler Aktivitäten.
Funktionale Anforderungen des mobilen Kontexts	
Vergleiche zu den Anforderungen 10-15 die Tabellen 8.2 und 8.3 sowie die Algorithmen 8.2 und 8.3	
Anforderung 10	Die Konzepte berücksichtigen den Ausführungsort.
Anforderung 11	Die Konzepte berücksichtigen den Netztyp.
Anforderung 12	Die Konzepte berücksichtigen den Formfaktor.
Anforderung 13	Die Konzepte berücksichtigen den Energiestatus.
Anforderung 14	Die Konzepte berücksichtigen die Dringlichkeit.
Anforderung 15	Die Konzepte berücksichtigen das Endanwenderverhalten.
Nicht-funktionale Anforderungen mobiler Aktivitäten	
Anforderung 16	Die Konzepte optimieren die Robustheit (vgl. Tabelle 8.1 und Algorithmus 8.1).
Anforderung 18	Die Konzepte ermöglichen ein frühzeitiges Erkennen von Laufzeitproblemen (vgl. die Algorithmen 8.2 und 8.3).

Tabelle 8.5: Beitrag zur robusten Ausführung mobiler Aktivitäten

8.6.2 Verwandte Ansätze

Verwandte Ansätze, die einen mobilen Kontext dazu nutzen, um die Ausführung mobiler Aktivitäten robuster zu gestalten, sind uns derzeit nicht bekannt. Auch die vorgestellte kontextbezogene mobile Endanwenderzuteilung wird von verwandten Ansätzen in der vorgestellten Form

nicht verfolgt.

Ansätze, die eine dynamische Ressourcenallokation in mobilen und ubiquitären Umgebungen betrachten [KAH⁺12], kommen dem hier vorgestellten Ansatz noch am nächsten. Dabei sind nur solche Ansätze relevant, die eine solche Allokation auch im Kontext eines *PrMS* betrachten. Eine Forschergruppe beschäftigt sich damit im Kontext nicht mobiler Aktivitäten [CRRC12b, RORC13, CRRC12a, CGR⁺13, CRRC11a, CKR⁺15, CRRC13, CRORC12, CKR⁺15]. Da dort kein mobiler Kontext betrachtet wird, lässt sich auch kein Vergleich anstellen.

Im mobilen Kontext berücksichtigen aktuell nur zwei Ansätze direkt die Möglichkeit, auf die Allokation von Aktivitäten zu Smart-Mobilgeräten Einfluss zu nehmen: [ZHKL10] diskutiert die verteilte Ausführung von Prozessen auf Basis der Instanz-Migration (vgl. Kapitel 3); Smart-Mobilgeräte werden als mögliche Ausführungsplattform mit eingeschlossen. Für jede Aktivität existiert in diesem Ansatz ein *Selection Type*, mit dem bestimmt werden kann, auf welchem Smart-Mobilgerät eine mobile Aktivität ausgeführt werden soll. Hier wird über weitere Selektionskriterien, die entweder ein *Algorithmus* bestimmt oder Kontextvariablen manuell festlegen, die mobile Umgebung dediziert berücksichtigt. Jedoch werden die Möglichkeiten der Selektion nur genannt und nicht weiter spezifiziert. Dabei geht es weniger um die kontextbezogene Zuteilung zu einem bestimmten mobilen Endanwender, sondern um eine flexible Möglichkeit, den gesamten Prozess von einem Smart-Mobilgerät zu einem anderen zu migrieren. [PCC⁺14] diskutiert einen mobilen Kontext und berücksichtigt auch die Fähigkeiten des Smart-Mobilgeräts. Eine Kostenrechnung berücksichtigt Aspekte wie Netzverbindung. Eine differenzierte Betrachtung des mobilen Kontexts, wie sie in dieser Arbeit erfolgt ist, gibt es in [PCC⁺14] nicht.

8.7 Zusammenfassung

In diesem Kapitel wurde gezeigt, in welcher Art und Weise der mobile Kontext verwendet wird, um die kontextbezogene mobile Endanwenderzuteilung für die Phasen der Modellierung, Erzeugung und Ausführung mobiler Aktivitäten zu realisieren. Außerdem wurde diskutiert, wie flexibel der mobile Kontext eingesetzt werden kann, um die robuste und kontextbezogene Ausführung mobiler Aktivitäten zu erzielen.

*Nichts ist entweder gut oder schlecht. Es sind unsere Gedanken,
die es dazu machen.*

Benjamin Franklin (1706-1790)

9

Mobile Aktivitäten und Ausnahmebehandlung

Gegenstand dieses Kapitels ist die Behandlung von Ausnahmen im Kontext mobiler Aktivitäten. Generell erfordert eine Ausnahmebehandlung während der Ausführung von Prozessen bzw. Aktivitäten aufwändige Analysen und Verfahren [WWLR06, RR06, GOR11, CLK99]. Allerdings betrachten nur wenige Arbeiten den Unterschied zwischen interaktiven und automatischen Aktivitäten im Zusammenhang mit Ausnahmebehandlungen. Ferner gibt es kaum Ansätze, die sich speziell mit Ausnahmen interaktiver mobiler Aktivitäten befassen.

Das in diesem Kapitel vorgestellte Verfahren zur Ausnahmebehandlung wird als *Ausnahmebehandlungsstrategie mobiler Aktivitäten* bezeichnet. Letztere bildet die Grundlage für eine robuste Ausführung mobiler Aktivitäten. Kapitel 9 gliedert sich wie folgt: Abschnitt 9.1 beschreibt die adressierten Ausnahmen, während Abschnitt 9.2 das entwickelte Verfahren zur Ausnahmebehandlung einführt. Da Letzteres drei Strategien umfasst, behandeln die folgenden Abschnitte 9.4, 9.5 und 9.6 diese Strategien im Detail. Abschnitt 9.7 diskutiert weitere Aspekte, während Abschnitt 9.8 sowohl den Beitrag dieses Kapitels zur robusten Ausführung mobiler Aktivitäten als auch verwandte Ansätze diskutiert. Abschnitt 9.9 fasst das Kapitel zusammen.

9.1 Einleitung

Welche Ausnahmen sind im Kontext interaktiver mobiler Aktivitäten von Interesse? Mobile Aktivitäten werden von einem Endanwender mit einem Smart-Mobilgerät ausgeführt. Daher liegt der Fokus auf Ausnahmen, die mit der Nichterreichbarkeit des Smart-Mobilgeräts durch das *PrMS* einhergehen. Andere Ausnahmen, etwa dass auf dem Smart-Mobilgerät ein Anwendungsprogramm, das durch den Programm-Klient gerufen wurde, nicht reagiert, werden nicht betrachtet. Hier können aber ähnliche Verfahren wie bei stationären Systemen zum Einsatz kommen.

Die Ausnahmen, die zu einer Nichterreichbarkeit des Smart-Mobilgeräts durch das *PrMS* führen, sind vielfältig. Ein Smart-Mobilgerät kann "crashen" (z.B. durch Herunterfallen), infolge eines niedrigen Batterieladestatus ausfallen, aufgrund eines Netzwechsels nicht erreichbar sein oder auch unbedacht von einem Endanwender ausgeschaltet worden sein. Bei Betrachtung solcher Ausnahmen wird ersichtlich, dass sich eine Ausnahmebehandlung für interaktive

mobile Aktivitäten anders gestaltet als im Fall stationärer Aktivitäten. Einerseits fallen Smart-Mobilgeräte mit einer höheren Wahrscheinlichkeit aus¹ als Desktop-Systeme, andererseits muss bei einem Desktop-System im Regelfall nicht einkalkuliert werden, dass ggf. ein Totalausfall des Systems vorliegt. Darüber hinaus ist es im Kontext mobiler Aktivitäten wichtig, dass die Ausnahmebehandlung den mobilen Kontext speziell berücksichtigt. So kann es vorkommen, dass ein mobiler Endanwender bewusst für längere Zeit Offline gehen will oder sogar muss, wenn es die Situation erfordert.

Generell soll die nachfolgend vorgestellte Ausnahmebehandlung weitgehend automatisch durchgeführt werden, d.h. manuelle Entscheidungen seitens der Endanwender sollen vermieden werden. Dies steigert sowohl die Robustheit als auch die Akzeptanz einer Ausnahmebehandlung. Hier ist zu beachten, dass ein mobiler Kontext für Endanwender mit Einschränkungen einhergeht. Diese sind für Endanwender oftmals nicht direkt ersichtlich. Beispielhaft sei ein automatischer Netzwechsel des Smart-Mobilgeräts genannt, der im weiteren Verlauf zu einer Offline-Situation führen kann.

9.2 Forschungsmethodik

Im Prozess-Management-Bereich werden unterschiedliche Verfahren der Ausnahmebehandlung betrachtet. Einen guten Überblick bieten [CLK99, RW12]. Zunächst fassen wir die drei grundsätzlichen Verfahren zur Ausnahmebehandlung zusammen und diskutieren, welche davon für mobile Aktivitäten geeignet sind.

- **Methode 1:** Die Ausnahme kann behoben werden, ohne die Ausführung des Prozesses bzw. der die Ausnahme auslösenden Aktivität zu verändern. So könnte ein Stellvertreter gesucht werden, der einen ausgefallenen mobilen Endanwender ersetzt.
- **Methode 2:** Die Ausnahme wird behoben, indem die Prozess- bzw. Aktivitätsausführung verändert wird. So kann die Aktivität übersprungen werden, falls dies der weitere Prozessablauf zulässt. Alternativ könnte eine andere Aktivität ausgewählt werden. Entsprechende Alternativen müssen allerdings vor Prozessbeginn festgelegt werden.
- **Methode 3:** Es wird eine Ad-hoc Adaption angewandt, d.h. zur Laufzeit wird das Schema eines Prozesses dynamisch geändert. Derartige Änderungen sind im Kontext einer Ausnahmebehandlung aufwändig und komplex [RW12].

In dieser Arbeit kommen die Methoden 1 und 2 zum Einsatz. Jedoch wurden grundlegende Änderungen im Vergleich zu [CLK99] vorgenommen. In [CLK99] werden diese beiden Methoden exklusiv gesehen, d.h. es kommt entweder die eine oder die andere Methode zum Einsatz. In dieser Arbeit werden die Methoden nach einem anderen Prinzip angewandt: Erst dann wenn Methode 1 nicht anwendbar ist, soll Methode 2 angewandt werden.

Diese Anpassung lässt sich mit dem mobilen Kontext begründen. Wenn eine aufgetretene Ausnahme keine Bedingungen des allgemeinen Prozesskontexts ändert, wird bei [CLK99] auch die Ausführung beibehalten.² Aus diesem Grund wurde Methode 1 zur Ausnahmebehandlung ausgewählt, da die Nichterreichbarkeit des Smart-Mobilgeräts keine neuen Bedingungen aufwirft und daher die ursprüngliche Ausführung beibehalten werden soll. Methode 2 wurde zusätzlich

¹Eine längerfristige Nichterreichbarkeit des Smart-Mobilgeräts ist aus Sicht des *PrMS* ein Ausfall.

²Beispiele solcher Bedingungen können in [CLK99] nachgelesen werden.

gewählt, da es in einem mobilen Kontext vorkommen kann, dass die ursprüngliche Ausführung nur sehr schwierig oder nur mit erheblicher Zeitverzögerung aufrecht erhalten werden kann. So kann der Ausfall eines WLAN dazu führen, dass kein mobiler Endwender die mobile Aktivität, in deren Kontext die Ausnahme aufgetreten ist, ausführen kann. In diesem Fall ermöglicht die Anwendung von Methode 2 eine gesicherte Behandlung der Ausnahme. Wichtig ist, dass Methode 2 erst zum Einsatz kommt, wenn Methode 1 nicht mehr durchführbar ist.

9.3 Ausnahmebehandlungsstrategie mobiler Aktivitäten

Abbildung 9.1 zeigt das entwickelte Verfahren zur Ausnahmebehandlung mobiler Aktivitäten unter Bezugnahme auf die Zustände mobiler Aktivitäten. Ferner wird die Reihenfolge der zur Anwendung gekommenen Methoden vorgestellt.

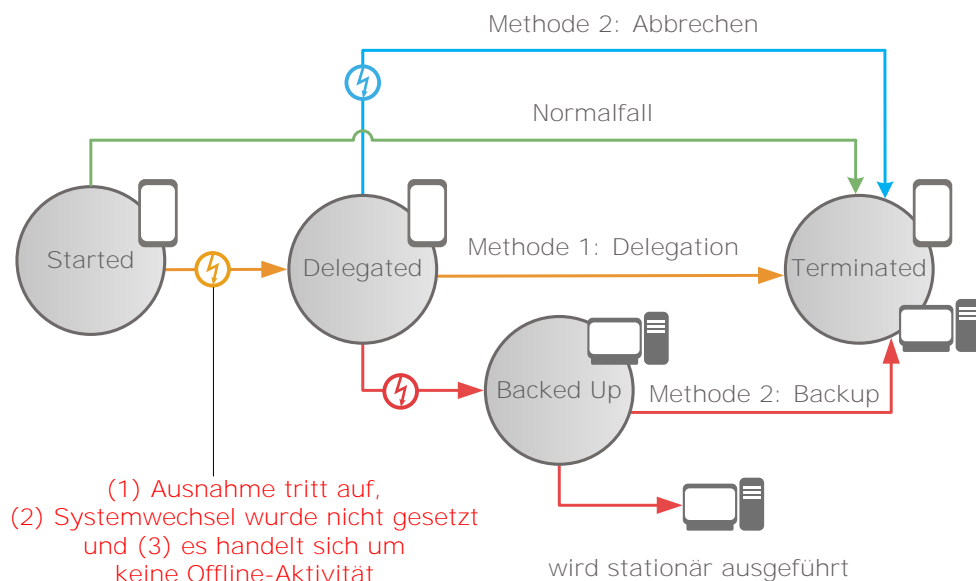


Abbildung 9.1: Ausnahmebehandlungsstrategie mobiler Aktivitäten

Sobald eine Ausnahme auftritt und der auslösende mobile Endanwender keinen Systemwechsel angefordert hat, sowie es sich um keine Offline-Aktivität handelt (vgl. Abb. 9.1), wird die Ausnahmebehandlung aktiviert. Ein Systemwechsel verhindert die Ausführung der Ausnahmebehandlung, da er grundsätzlich sicherstellt (vgl. Kapitel 8), dass die mobile Aktivität ordnungsgemäß ausgeführt werden kann. Aus diesem Grund ist bei einem angeforderten Systemwechsel keine Ausnahmebehandlung notwendig. Bei der Offline-Aktivität wird die Ausnahmebehandlung wiederum in angepasster Form angewandt, da die Offline-Situation gezielt durch einen mobilen Endanwender gewählt wurde.

Der erste grundlegende Schritt der Ausnahmebehandlung besteht in der Durchführung einer sog. Delegation. Damit ist gemeint, dass die Ausführung der mobilen Aktivität durch das *PrMS* an einen anderen, ebenfalls geeigneten mobilen Endanwender übergeben wird. Weiters geschieht diese Übertragung in automatischer Art und Weise. Dazu muss das *PrMS* insbesondere prüfen, welche mobilen Endanwender zur Delegation geeignet sind.

Es kann auch vorkommen, dass gar keine Delegation möglich ist. Dieser Fall tritt ein, wenn kein mobiler Endanwender mehr zur Verfügung steht, der zur Ausführung der mobilen Aktivität berechtigt ist. In diesem Fall geht die Ausnahmebehandlung folgendermaßen vor: Es wird überprüft, ob es sich um eine kritische oder unkritische mobile Aktivität handelt. Ersteres bedeutet, dass der weitere Prozessablauf vom Ergebnis der mobilen Aktivität abhängt, etwa wenn diese Daten erzeugt, die von nachfolgenden Aktivitäten gelesen werden. Bei unkritischen mobilen Aktivitäten hängt der weitere Prozessverlauf nicht von dieser mobilen Aktivität ab. Wurde eine kritische mobile Aktivität identifiziert, wird ein sog. Backup angewandt. Dieses überführt die Ausführung der mobilen Aktivität auf ein Desktop-System, sodass eine gesicherte Ausführung gewährleistet werden kann. Im Fall einer unkritischen mobilen Aktivität, wird diese abgebrochen, d.h. ihre Ausführung wird übersprungen.

Um die Ausnahmebehandlung umzusetzen, muss das in Kapitel 5 vorgestellte Zustandsmodell für mobile Aktivitäten entsprechend erweitert werden. Abbildung 9.2 zeigt das erweiterte Zustandsmodell. Es werden zwei neue Zustände für die Delegation und das Backup eingeführt. Außerdem werden die bestehende Zustandsübergänge angepasst. Die Anpassungen für die Zustandsübergänge betreffen einerseits die neuen Zustände, andererseits werden auch für bestehende Zustände neue Zustandsübergänge eingeführt. So muss beispielsweise ein neuer Zustandsübergänge zwischen den bestehenden Zuständen *Started* und *Selected* eingeführt werden.

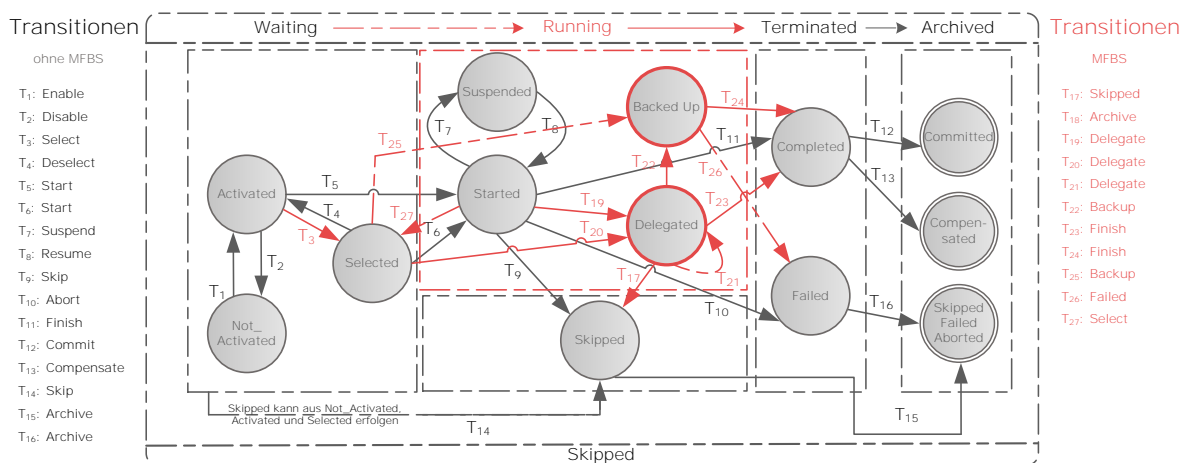


Abbildung 9.2: Erweitertes Zustandsmodell (vereinfacht)

9.4 Delegation mobiler Aktivitäten

Die primäre Strategie der Ausnahmebehandlung besteht in der Delegation mobiler Aktivitäten. Tritt eine zu behandelnde Ausnahme³ auf, wird bei einer Delegation die Ausführung der mobilen Aktivität automatisch an einen anderen geeigneten mobilen Endanwender übertragen. Damit einhergehend wird dem Endanwender, dessen Smart-Mobilgerät die Ausnahme verursacht hat, die Ausführung entzogen. An dieser Stelle sei erwähnt, dass es sich um eine Ausführungsdelega-

³Diese Ausnahme betrifft die Nichterreichbarkeit des mobilen Endanwenders durch das *PrMS*, sie wird in Tabelle 9.2 konkretisiert.

tion handelt, d.h. die Ausführung der mobilen Aktivität wird an einen anderen mobilen Endanwender übertragen. Dem gegenüber steht die Rechtedelegation, bei der Endanwendern temporär Ausführungsrechte anderer Endanwender eingeräumt werden [WKB07, AW05, GSFC08]. Bei einer Ausführungsdelegation sind verschiedene Aspekte zu berücksichtigen. Die wichtigsten fasst Tabelle 9.1 zusammen.

	Delegationsaspekte	Erfordert
DA1	Welche konkreten Ausnahmen müssen seitens des <i>PrMS</i> unterschieden werden, um eine Delegation zu starten?	Betrachtung des mobilen Kontexts.
DA2	Welche mobilen Endanwender eignen sich als Subjekt (Zielsubjekt) einer Delegation?	Betrachtung des mobilen Kontexts und des Ranking-Algorithmus.
DA3	Wie wird eine Delegation grundsätzlich durchgeführt?	Betrachtung des Ranking-Algorithmus.
DA4	Welche weiteren Aspekte sind für den mobilen Endanwender bei der Durchführung der Delegation zu berücksichtigen?	Betrachtung des Prozess-Klienten.
DA5	Welche weiteren Aspekte sind bezogen auf das <i>PrMS</i> bei der Durchführung der Delegation zu berücksichtigen?	Betrachtung des Ausführungsprotokolls.
DA6	Können aus praktischer Sicht mehrere Delegationsvarianten unterschieden werden?	

Tabelle 9.1: Fragestellungen im Kontext der Delegation mobiler Aktivitäten

9.4.1 Fragestellung DA1: Starten einer Delegation

Wann eine Delegation gestartet werden soll, ergibt sich aus dem mobilen Kontext und dessen Parametern zur Abfrage des Online-Status eines Smart-Mobilgeräts (d.h. MA_{SFOS} und MA_{AZOS}). Als weiterer Parameter wird *atimeout* (vgl. Tabelle 9.2) eingeführt. Er gibt die Zeit an, innerhalb der eine Rückmeldung auf Basis von MA_{SFOS} oder MA_{AZOS} erfolgen muss. Erfolgt diese Rückmeldung nicht, wird eine Delegation angestoßen, allerdings nur wenn es sich um keine Offline-Aktivität handelt.

Schließlich kann auch der Kontextparameter zur Dringlichkeit dazu führen, dass eine Delegation angestoßen wird. Dies ist der Fall, wenn die Zeitdauer, die durch die Dringlichkeit festgelegt wird, abgelaufen ist. Dabei kann es vorkommen, dass der mobile Endanwender weiter Online ist. Für diesen Fall ist eine Delegation in der Regel kontraproduktiv. Daher wird bei einer abgelaufenen Dringlichkeit eine Nachricht an das Smart-Mobilgerät des mobilen Endanwenders geschickt, um den mobilen Endanwender aufzufordern, entweder die Dringlichkeit zu bestätigen und die mobile Aktivität zu delegieren oder aber die Ausführung zu behalten. Letzterer Fall kann jedoch im weiteren Verlauf dazu führen, dass der mobile Endanwender lange für die Ausführung der mobilen Aktivität benötigt. Daher wird für diesen Fall ein Timer mit Zeitdauer *atimeout* gestartet. Läuft dieser ab, wird automatisch eine Delegation durchgeführt und der mobile Endanwender bekommt die Ausführung endgültig entzogen.

Einführung zusätzlicher Parameter	
<i>atimeout</i>	Der Parameter <i>atimeout</i> wird für einen gesamten Prozess festgelegt. Er gibt die Zeitspanne an, innerhalb der keine Ausnahmebehandlung stattfinden soll.
Parametervergleich	Beschreibung
$MA_{SFOS} > 0 \wedge MA_{OFF} = false$	Wenn innerhalb der Zeitspanne <i>atimeout</i> keine Rückmeldung des Smart-Mobilgeräts auf Basis von MA_{SFOS} erfolgte, wird eine Delegation angestoßen.
$MA_{AZOS} > 0 \wedge MA_{OFF} = false$	Wenn innerhalb der Zeitspanne <i>atimeout</i> keine Rückmeldung des Smart-Mobilgeräts auf Basis von MA_{AZOS} erfolgte, wird eine Delegation angestoßen.
$MA_D = atimeout$	Wenn eine Dringlichkeit fixiert wurde und diese <i>atimeout</i> beträgt, wird ebenfalls eine Delegation angestoßen.

Tabelle 9.2: Kontextregeln zur Delegationsaktivierung

9.4.2 Fragestellung DA2: Ermittlung geeigneter mobiler Endanwender

Generell werden die für mobile Aktivitäten geeigneten mobilen Endanwender durch einen Ranking-Algorithmus bestimmt (vgl. Kapitel 8). Dieser wird im Folgenden angepasst, um die Bestimmung geeigneter mobiler Endanwender auch für den Fall einer Delegation zu optimieren. Dies wird erforderlich, da eine Delegation die Behandlung einer Ausnahme darstellt; d.h. die Bestimmung geeigneter mobiler Endanwender sollte diese spezielle Situation möglichst gut berücksichtigen. Hier stellt sich die Frage, ob weitere Parameter gefunden werden können, um diese Situation spezifisch zu berücksichtigen.

Tabelle 9.3 zeigt Anpassungen der Kontextregeln für den Ranking-Algorithmus aus Kapitel 8. Diese Anpassungen berücksichtigen insbesondere den mobilen Kontext für eine Delegation. Entscheidend sind zwei Aspekte: Erstens führen die Kontextregeln vier neue Parameter (ME_{CL} , ME_{LBS} , ME_{RB} , ME_{DB}) ein, die das Verhalten eines mobilen Endanwender bezogen auf die Ausnahmebehandlung erfassen. So beschreibt der Parameter ME_{DB} (vgl. Tabelle 9.3), ob für den mobilen Endanwender häufig Delegationen notwendig werden. Zweitens ermöglicht der Parameter ME_{DPV} eine Priorisierung der verfügbaren mobilen Endanwender zur Durchführung der Delegation. Mithilfe dieser Kontextregeln wird der Ranking-Algorithmus aus Kapitel 8 für den Zweck einer Delegation entsprechend angepasst. Der resultierende Algorithmus 9.1 berechnet den Ranking-Parameter ME_{DPV} für alle geeigneten mobilen Endanwender zur Delegation einer mobilen Aktivität.

9.4.3 Fragestellung DA3: Delegationsdurchführung

Auf Grundlage der von Algorithmus 9.1 priorisierten Liste $KMEZL(n)$ wird schließlich eine Delegation durchgeführt. Dazu wird das erste Element aus der Liste $KMEZL(n)$ ausgewählt und dem betreffenden mobilen Endanwender wird die mobile Aktivität zuerst zur Delegation angeboten. D.h. dieser mobile Endanwender wird aktuell als am besten geeignet für die Delegation angesehen, da sein Ranking-Wert ME_{DPV} den besten Wert aufweist. Weiters ist bei einer Delegation entscheidend, dass diese immer nur einem mobilen Endanwender angeboten wird, d.h. nicht allen mobilen Endanwendern aus $KMEZL(n)$ der mobilen Aktivität n . Dies dient

Einführung zusätzlicher Parameter		
<p>ME_{CL}: Dieser Parameter gibt an, wie oft ein mobiler Endanwender <i>atimeout</i> überschritten hat (CL: Connection Losses).</p> <p>ME_{LBS}: Dieser Parameter ist ein Zähler. Er wird für einen mobilen Endanwender immer dann erhöht, wenn beim Start einer mobilen Aktivität $SMG_{BLS} < M_{ANBLS}$ zutrifft (LBS: Low Battery Time).</p> <p>ME_{RB}: Dieser Parameter fasst die Parameter ME_{CL} und ME_{LBS} zusammen und beschreibt das Ressourcenverhalten eines mobilen Endanwenders (Resource Behaviour).</p> <p>ME_{DB}: Dieser Parameter bildet für einen mobilen Endanwender das Verhältnis zwischen angestoßenen Delegationen und den insgesamt von ihm gestarteten mobilen Aktivitäten. Der Parameter beschreibt das Delegationsverhalten eines mobilen Endanwenders (Delegation Behaviour).</p> <p>ME_{DPV}: Dieser Parameter beschreibt für einen mobilen Endanwender, inwieweit er für eine Delegation geeignet ist (Delegation Priorization Value).</p>		
Parametervergleich	Ranking	PrMS
$0 < normEZB(ME) \leq 1$	✓	X
$MA_{SAO} = ME_{SAO}$	✓	X
$SMG_{BLS} < M_{ANBLS}$	✓	X
$ME_{ISZ} > 0$	✓	X
$ME_{RB} > 0$	✓	X
$ME_{DB} > 0$	✓	X
Formeln		
$normEZB(ME) = \begin{cases} \frac{MA_{GAO} - SMG_{GAO}(ME)}{MA_{EB}}, & \text{wenn geometrische Daten bestimmbar} \\ 0, & \text{wenn geometrische Daten nicht bestimmbar} \end{cases} \quad (9.1)$		
$ME_{RB} = \begin{cases} ME_{CL} + ME_{LBS}, & ME_{CL} \neq 0 \\ 0, & ME_{CL} = 0 \end{cases} \quad (9.2)$		
$ME_{DB} = \begin{cases} \frac{\#Aktivitäten \text{ von } ME \text{ delegiert}}{\#Aktivitäten \text{ von } ME \text{ gestartet}}, & \text{wenn Akt. von } ME \text{ delegiert} \neq 0 \\ 0, & \text{wenn Akt. von } ME \text{ delegiert} = 0 \end{cases} \quad (9.3)$		
Funktionale Abhängigkeiten		
$SMG_{ONT} \rightarrow SMG_{BLS}$	X	✓
$MA_{GAO} \rightarrow MA_{EB}$	X	✓
X =trifft nicht zu, ✓=trifft zu		

Tabelle 9.3: Kontextregeln für Delegationen

der robusten Ausführung mobiler Aktivitäten.

Da bei einer Delegation eine Ausnahme vorliegt und die mobile Aktivität bereits ausgeführt worden ist, sollte nicht mehr vordergründig sein, welcher mobile Endanwender die Aktivität ausführen will, sondern der am besten geeignetste mobile Endanwender sollte möglichst schnell gefunden werden. Daher wird nur ein mobiler Endanwender benachrichtigt. Sollte dieser die Delegation ablehnen, wird die zuvor ermittelte Liste $KMEZL(n)$ bezüglich ME_{DPV} so lange durchlaufen, bis ein mobiler Endanwender gefunden wird, der die Delegation annimmt.⁴

Abschließend sei erwähnt, dass auch eine Delegation eine gewisse Zeit in Anspruch nimmt. Währenddessen könnte der die Ausnahme verursachende mobile Endanwender wieder Online gegangen sein. Deshalb wird als weitere Maßnahme vor jedem Delegationsversuch aus $KMEZL(n)$ nochmals überprüft, ob der die Ausnahme verursachende mobile Endanwender inzwischen wieder Online ist.

⁴Eine weitere Maßnahme zur Erweiterung von $KMEZL(n)$ wäre eine Delegation von Rechten durchzuführen (vgl. Kapitel 8)

Algorithmus 9.1 : Bestimmung von $KMEZL(n)$ bei Delegationen

Data : Relevante Parameterwerte des mobilen Kontexts (vgl. Tabelle 7.1)

Regeln des mobilen Kontexts (vgl. Tabelle 8.2)

 $MAME \subseteq PEA(n)$: Menge angemeldeter und berechtigter mobiler Endanwender der mobilen Aktivität n
Result : $KMEZL(n)$: Liste geeigneter mobiler Endanwender zur Delegation der mobilen Aktivität n

```

1  begin
2       $KMEZL(n) \leftarrow \emptyset$ ;          /* Liste geeigneter mobiler Endanwender initialisieren */
3      /* Für mobile Endanwender aus MAME  $ME_{DPV}$  berechnen und in  $KMEZL(n)$  einfügen */
4       $ea_i \leftarrow \emptyset$ 
5      foreach Endanwender  $ea \in MAME$  do
6          switch  $0 < normEZB(ME(ea)) \leq 1 \vee MA_{SAO}(n) = ME_{SAO}(ea)$  do
7              /* Die beste Priorisierung  $ME_{DPV}$  wird durch eine geometrische Ortsübereinstimmung
8              erreicht */
9              case  $0 < normEZB(ME(ea)) \leq 1$ 
10                  $ME_{DPV}(ea)[ea_i] \leftarrow 0 + ME_{DB}(ea) + ME_{RB}(ea) + ME_{ISZ}(ea)$ ;
11                  $KMEZL(n) \leftarrow KMEZL(n) \cup \{ea\}$ ;
12             /* Die nächstbeste Priorisierung  $ME_{DPV}$  wird durch eine symbolische
13             Ortsübereinstimmung erreicht */
14             case  $MA_{SAO}(n) = ME_{SAO}(ea)$ 
15                  $ME_{DPV}(ea)[ea_i] \leftarrow 1 + ME_{DB}(ea) + ME_{RB}(ea) + ME_{ISZ}(ea)$ ;
16                  $KMEZL(n) \leftarrow KMEZL(n) \cup \{ea\}$ ;
17             /* Dieser Fall deckt die übrigen Fälle ab, in denen keine Ortsübereinstimmung
18             erreicht werden kann */
19             otherwise
20                  $ME_{DPV}(ea)[ea_i] \leftarrow 2 + ME_{DB}(ea) + ME_{RB}(ea) + ME_{ISZ}(ea)$ ;
21                  $KMEZL(n) \leftarrow KMEZL(n) \cup \{ea\}$ ;
22             end
23         endsw
24          $ea_i ++$ ;
25     end
26     /*  $KMEZL(n), KMEZL(n)[1], \dots, KMEZL(n)[m]$  gemäß  $ME_{DPV}$  aufsteigend sortieren */
27      $tmp \leftarrow \emptyset$ ;          /* TMP-Variable für einzusortierenden Wert initialisieren */
28      $tmp2 \leftarrow \emptyset$ ;      /* TMP-Variable für einzusortierenden Wert initialisieren */
29      $j \leftarrow 0$ ;              /* Hilfszähler initialisieren */
30     for  $i \leftarrow 1$  to  $|KMEZL(n)|$  do
31          $tmp \leftarrow KMEZL(n)[i]$ ;
32          $tmp2 \leftarrow ME_{DPV}(ea)[i]$ ;
33          $j \leftarrow i$ ;
34         while  $(j > 1 \wedge KMEZL(n)[j-1] > ME_{DPV}(ea)[i])$  do
35              $KMEZL(n)[j] \leftarrow KMEZL(n)[j-1]$ ;
36              $ME_{DPV}(ea)[j] \leftarrow ME_{DPV}(ea)[j-1]$ ;
37              $j \leftarrow j-1$ ;
38              $KMEZL(n)[j] \leftarrow tmp$ ;
39              $ME_{DPV}(ea)[j] \leftarrow tmp2$ ;
40         end
41     end
42 end
    
```

9.4.4 Fragestellung DA4: Wichtige Aspekte für den Endanwender

In welcher Art und Weise wirkt sich nun eine Delegation für den die Ausnahme verursachenden mobilen Endanwender aus? Und wie stellt sich die Ausnahme für den Endanwender, der die

Delegation angeboten bekommt, dar? Prinzipiell sind für den die Ausnahme verursachenden mobilen Endanwender zwei Fälle zu unterscheiden (vgl. Tabelle 9.4).

Beschreibung	
F1	Das Smart-Mobilgerät des mobilen Endanwenders ist nicht mehr funktionstüchtig. Beispielsweise kann die Batterie leer sein, das Gerät unbedacht ausgeschaltet worden sein oder ein Crash des Geräts vorliegen.
F2	Das Smart-Mobilgerät des mobilen Endanwenders funktioniert weiterhin, kann aber durch das <i>PrMS</i> nicht erreicht werden. So kann sich das Smart-Mobilgerät in einem Netz befinden, das nicht vom <i>PrMS</i> erreicht werden kann.

Tabelle 9.4: Fallunterscheidung Smart-Mobilgerät

Für beide Fälle **F1** und **F2** sind zwei Phasen relevant. Die erste Phase beginnt mit dem Eintreten der Ausnahme. Die zweite Phase wiederum beginnt sobald das Smart-Mobilgerät wieder eine Verbindung zum *PrMS* hat. Die Umsetzung der ersten Phase unterscheidet sich für die Fälle **F1** und **F2**, während die zweite Phase identisch durchgeführt wird.

In der ersten Phase ist für Fall **F1** nichts weiter zu betrachten, da das Smart-Mobilgerät nicht funktioniert. Die erste Phase für Fall **F2** unterscheidet sich davon grundlegend, da der mobile Endanwender weiterarbeiten kann, sein Smart-Mobilgerät hat lediglich keine Verbindung zum *PrMS*. Das Weiterarbeiten muss dennoch verhindert werden, da die gerade ausgeführte mobile Aktivität delegiert wird und es so passieren kann, dass zwei mobile Endanwender parallel an derselben mobilen Aktivität arbeiten. Um diesen Fall zu vermeiden, wurde ein neuer Mechanismus für den mobilen Prozess-Klienten entwickelt, der nachfolgend beschrieben wird.

Eine Voraussetzung zur Behandlung von Phase 1 ist ferner, dass das Smart-Mobilgerät die Ausnahme ebenfalls erkennt. Dazu wird folgender Mechanismus angewandt: Beim Starten der Aktivität wird dem Smart-Mobilgerät der Parameter *atimeout* (vgl. Tabelle 9.3) vom *PrMS* mitgeteilt. Nach jeder Online-Rückmeldung an das *PrMS* startet das Smart-Mobilgerät diesen Timer. Kann innerhalb des Timers keine Rückmeldung erfolgen, wird der Zustand einer Ausnahme lokal festgestellt.

Die Behandlung von Phase 1 zeigt Abbildung 9.3.

Sobald die Ausnahme lokal (d.h. auf dem Smart-Mobilgerät des schaffenden mobilen Endanwenders) festgestellt worden ist, werden alle sich in Ausführung befindlichen Aktivitäten in den Zustand *In-Delegation* versetzt (vgl. Abb. 9.3①). Ferner werden alle anderen Aktivitäten, die dieser mobile Endanwender angenommen, bislang aber nicht gestartet hat, in den Zustand *In-Delegation* versetzt (vgl. Abb. 9.3②). Dieser Zustand drückt aus, dass diese Aktivitäten nun lokal nicht mehr bearbeitet werden können. Damit ist Phase 1 abgeschlossen. Phase 1 stellt somit insbesondere sicher, dass mobile Aktivitäten im Zustand der Ausnahme nicht bearbeitet werden können.

Phase 2 wird für die Fälle **F1** und **F2** (vgl. Tabelle 9.4) in gleicher Art und Weise durchgeführt (vgl. Abb. 9.3 rechts). Stellt der mobile Endanwender die Verbindung zum *PrMS* wieder her, fragt sein Smart-Mobilgerät den Status der mobilen Aktivitäten an, die zuletzt in der Worklist gespeichert waren. Der Fall, dass diese Information nicht mehr vorliegt, wird nicht weiter diskutiert. Es gibt nur wenige Fälle, in denen man diese Information lokal nicht dauerhaft sichern kann, das schließt sogar die meisten Geräte-Crashes mit ein. Das *PrMS* sendet nun für die letzten Worklist-Einträge vor der Ausnahme den aktuellen Status an das Smart-Mobilgerät. Es können nun für jeden Worklist-Eintrag zwei Fälle eintreten. Entweder wurde die mobile



Abbildung 9.3: Prozess-Klient mit verursachter Delegation

Aktivität, die den Worklist-Eintrag repräsentiert, an einen anderen mobilen Endanwender delegiert (vgl. Abb. 9.3③) oder die Delegation konnte noch nicht durchgeführt werden (vgl. Abb. 9.3④). Im letztgenannten Fall kann der mobile Endanwender die mobile Aktivität wieder zurückfordern. Führt er die Rückforderung durch, wird ihm die Aktivität sofort wieder zugeteilt. Dies ist jedoch nur dann möglich, wenn die Ausnahmebehandlung nicht Methode 2 (Abbruch bzw. Backup) der Ausnahmebehandlung angestoßen hat (vgl. Abschnitt 9.2). Damit ist Phase 2 ebenfalls abgeschlossen.

Als nächstes wird vorgestellt, wie sich die Delegation für denjenigen mobilen Endanwender auswirkt, der die Delegation angeboten bekommt (vgl. Abb. 9.4). Wie in Abbildung 9.4① zu sehen ist, wird dem mobilen Endanwender die Delegationsanfrage in seiner Worklist angezeigt. Außerdem erkennt er am Status, dass es sich um eine Delegation handelt (vgl. Abb. 9.4②). Weiters wird angezeigt, wer der die Ausnahme verursachende mobile Endanwender ist (vgl. Abb. 9.4③). Der mobile Endanwender, der die Delegation erhalten hat, muss nun entscheiden, ob er sie annimmt oder ablehnt (vgl. Abb. 9.4④). Nimmt er die Delegation an, wird die Aktivität in seine Worklist übertragen. Lehnt er sie ab, wird der nächste mobile Endanwender in $KMEZL(n)$ ausgewählt und die Delegationsanfrage an ihn geschickt.

9.4.5 Fragestellung DA5: Wichtige Aspekte für das PrMS

Dieser Abschnitt stellt das Ausführungsprotokoll mobiler Aktivitäten im Kontext einer Delegation vor. Das Standard-Ausführungsprotokoll mobiler Aktivitäten wurde in Kapitel 8 behandelt. Die Delegation erfordert nun eine Anpassung dieses Protokolls.

Der Bereich des Ausführungsprotokolls, der von einer Ausnahme betroffen ist, wird in Abbildung 9.5 dargestellt (Schritte 4-15). Die zu einer Delegation führende Ausnahme tritt zeitlich

Aktivität	Status	Zeitlimit	Delegation von	Annahme?
Blutabnahme Frau Heller	DELEGATION	4h 30min	Johannes Schobel	<input type="radio"/> Ja <input type="radio"/> Nein

(a) Marc Schickler:
Delegations-Anfrage**Abbildung 9.4:** Prozess-Klient mit angebotener Delegation

nach Schritt 4 des Ausführungsprotokolls aus, da sich ab diesem Zeitpunkt die Aktivität in Ausführung befindet. Schritt 4 sorgt dafür, dass die Aktivität lokal auf dem Smart-Mobilgerät gestartet wird. Schritt 15 sorgt schließlich dafür, dass die ordnungsgemäße Ausführung und das Ausführungsergebnis vom Smart-Mobilgerät an das *PrMS* übertragen werden. Zwischen den Schritten 4 und 15 kann schließlich die zu einer Delegation führende Ausnahme auftreten.

Das Ausführungsprotokoll für die Delegation zeigt Abbildung 9.6. Weiters diskutiert Tabelle 9.5 die einzelnen Schritte aus Abbildung 9.6 im Detail. Hier sei angemerkt, dass die Ausnahme in Abbildung 9.6 nach Schritt 6 auftritt. Dies stellt nur eine Möglichkeit dar. Die Ausnahme könnte auch bereits vor Schritt 6 erfolgen. Letzteres bedeutet, dass die Protokollschritte ab Auftreten der Ausnahme bis Schritt 8 nicht durchgeführt werden. Da das Auftreten einer Ausnahme nach Schritt 6 bedeutet, dass alle möglichen Protokollschritte durchzuführen sind, wird dieser Fall nachfolgend kurz vorgestellt.

Drei Aspekte seien abschließend für das Ausführungsprotokoll einer Delegation erwähnt.

- Stellt das *PrMS* bei Schritt 2 fest (vgl. Abb. 9.6②), dass das Smart-Mobilgerät wieder Online ist, wird die Delegation abgebrochen.
- Die Propagierung von Daten, die während der Delegation auf dem Smart-Mobilgerät zwischengespeichert wurden und die noch nicht zum *PrMS* übertragen worden sind, erhöht die robuste Ausführung mobiler Aktivitäten ebenfalls. Dadurch können mobile Endanwender, die eine Delegation durchführen, evtl. bereits erzeugte Daten wiederverwenden.
- Die Protokolländerungen führen dazu, dass die Implementierung (vgl. Abb. 9.6⑥) des Prozess-Klienten auf dem Smart-Mobilgerät entsprechend geändert wird.

Ein weiterer Aspekt zur robusten Gestaltung mobiler Aktivitäten ist die Zwischenspeicherung der vom Smart-Mobilgerät erzeugten Daten. Diese Zwischenspeicherung bedeutet, dass das

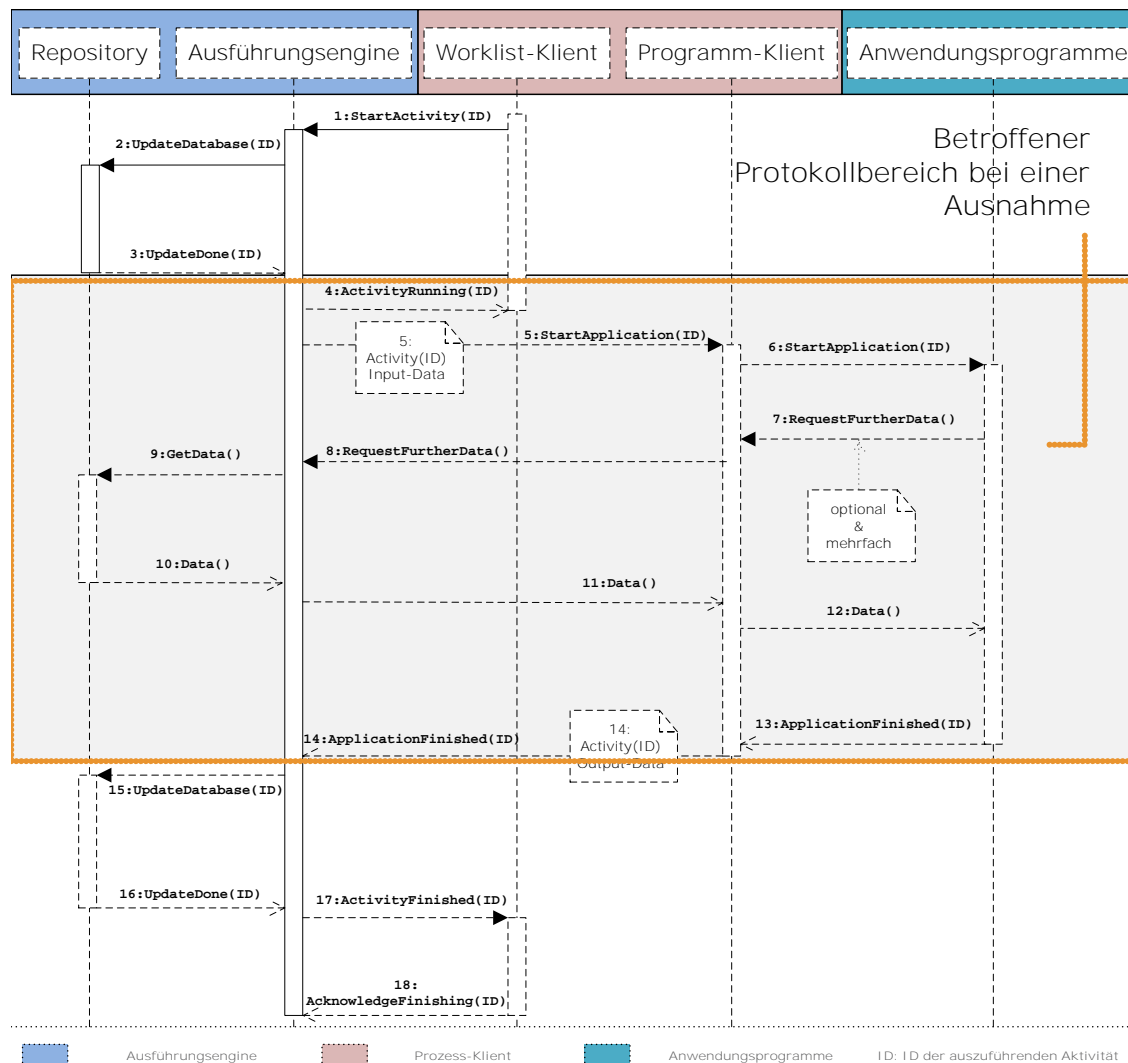


Abbildung 9.5: Betroffener Bereich des Ausführungsprotokolls bei Ausnahmen

Smart-Mobilgerät auch während der Ausführung einer mobilen Aktivität, zugehörige Zwischenergebnisse an das *PrMS* überträgt. Hintergrund dieser Zwischenspeicherung ist, dass bei einer Delegation die zwischengespeicherten Ergebnisse vom *PrMS* an den Endanwender, der die Delegation annimmt, übertragen werden können. Auf diese Weise gehen möglichst wenig Daten verloren, was die robuste Ausführung mobiler Aktivitäten erhöht. Dazu sind entsprechende Änderungen des Ausführungsprotokolls notwendig (vgl. Abb. 9.7①).

Für das Zwischenspeichern von Ergebnissen sind die Schritte 7 bis 14 in Abbildung 9.7 relevant. Tabelle 9.6 stellt diese im Detail vor.

Zuletzt seien drei Aspekte bezogen auf das Ausführungsprotokoll bei der Zwischenspeicherung von Daten erwähnt. Erstens ist die Zwischenspeicherung nur möglich, wenn das Anwendungsprogramm eine solche Abfrage von Zwischenergebnissen zulässt (vgl. Abb. 9.7②). Zweitens erfordert diese Änderung des Ausführungsprotokolls ebenfalls wieder eine entsprechende Ände-

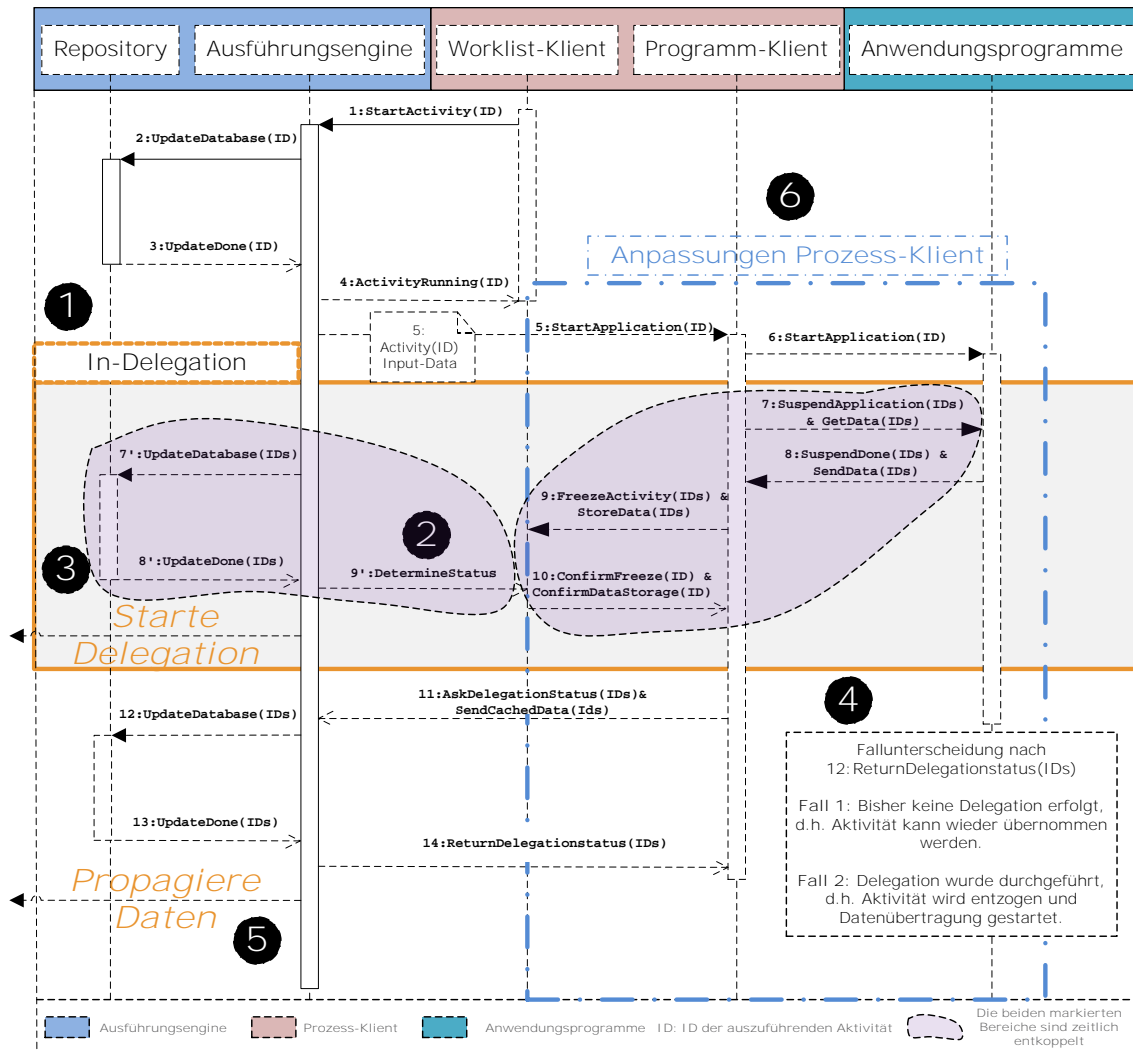


Abbildung 9.6: Ausführungsprotokoll und Delegationen

rung der Implementierung des Prozess-Klienten (vgl. Abb. 9.7③). Drittens wird die Zwischenspeicherung von Daten auch für den Systemwechsel angewandt (vgl. Kapitel 8).

9.4.6 Fragestellung DA6: Delegationsvarianten

Tabelle 9.9 stellt die beiden zur Anwendung kommenden Delegationsvarianten **DV1** und **DV2**⁵ vor. Die bisherigen Ausführungen beziehen sich auf eine Delegation, die genau dann durchgeführt wird, wenn eine Ausnahme auftritt, die sich aus der Nichterreichbarkeit des Smart-Mobilgeräts ergibt (vgl. Tabelle 9.9DV1). Es kann zudem vorkommen, dass mehr als eine Delegation bezogen auf dieselbe mobile Aktivität erfolgt. Generell sind Delegationen möglich, solange sich noch mobile Endanwender in der Liste $KMEZZ(n)$ befinden (vgl. Abschnitt 9.4.2).

⁵DV2 besteht aus den Subvarianten DV2-1 und DV2-2.

Phase	Beschreibung	PK	PrMS
1 – 6	Die Phasen 1-6 werden im Vergleich zum bisherigen Ausführungsprotokoll nicht verändert.	✓	✓
Ausnahme tritt auf (vgl. Abb. 9.6①)			
7 u. 8	Alle lokal sich in Ausführung befindlichen mobilen Aktivitäten werden vom Programm-Klienten suspendiert, der zudem die erzeugten Daten von den lokalen Anwendungsprogrammen anfordert.	✓	X
7' u. 8'	Das <i>PrMS</i> ändert den Status, der an den Programm-Klienten zugeteilten mobilen Aktivitäten.	X	✓
9 u. 10	Die suspendierten mobilen Aktivitäten werden vom Programm-Klienten eingefroren und die erzeugten Daten zwischengespeichert.	✓	X
9'	Das <i>PrMS</i> fragt nochmals beim Programm-Klient nach, ob das Smart-Mobilgerät wieder Online ist (vgl. Abb. 9.6②).	X	✓
Das <i>PrMS</i> startet die Delegation (vgl. Abb. 9.6③)			
11	Das Smart-Mobilgerät hat wieder Verbindung zum <i>PrMS</i> und der Programm-Klient ruft den Zustand der Aktivitäten seiner Worklist beim <i>PrMS</i> ab. Weiters schickt er noch nicht propagierte Daten an das <i>PrMS</i> .	✓	X
14	Das <i>PrMS</i> meldet dem Programm-Klienten den Status der angeforderten Aktivitäten. Ferner werden auf dem Smart-Mobilgerät zwischengespeicherte Daten, die noch nicht an das <i>PrMS</i> übertragen wurden, nun an das <i>PrMS</i> übertragen. Diese werden in der Folge vom <i>PrMS</i> an diejenigen Programm-Klienten geschickt, die eine Delegation erhalten und noch nicht abgeschlossen haben.	X	✓
Das Smart-Mobilgerät kann Aktivitäten zurückfordern, die nicht delegiert wurden. Delegierte Aktivitäten werden aus der Worklist entfernt (vgl. Abb. 9.6④).			
<p style="text-align: center;"> <i>X</i> = trifft nicht zu, <i>✓</i> = trifft zu <i>PK</i> = Prozess-Klient, <i>PrMS</i> = Prozess-Management-System Die "''"-Markierung bei den Phasen zeigt an, dass dort eine zeitliche Entkopplung zwischen <i>PrMS</i> und Prozess-Klient besteht. </p>			

Tabelle 9.5: Phasen des Ausführungsprotokolls bei Delegationen

Wie Tabelle 9.9 zu entnehmen ist, wird zusätzlich das Konzept der *erzwungenen Delegation* eingeführt. Letzteres bedeutet, dass der mobile Endanwender, an den delegiert werden soll, nicht mehr explizit benachrichtigt wird, sondern die betreffende mobile Aktivität wird ihm automatisch in seine Worklist eingestellt. Für die erzwungene Delegation werden ferner zwei Untervarianten **DV2-1** und **DV2-2** unterschieden (vgl. Tabelle 9.9).

Untervariante **DV2-1** der erzwungenen Delegation wird bei folgendem Szenario angewandt. In Abschnitt 9.4.2 wurde ausgeführt, dass die Liste $KMEZZL(n)$ durchlaufen wird, um einen geeigneten mobilen Endanwender für die Delegation zu finden. Dazu wird immer nur ein mobiler Endanwender angefragt, bis im Extremfall alle mobilen Endanwender aus $KMEZZL(n)$ abgelehnt haben. Tritt dieser Fall ein, wird eine erzwungene Delegation durchgeführt. Dazu wird die Liste $KMEZZL(n)$ abermals bestimmt, da sich zwischenzeitlich Parameter der mobilen Endanwender geändert haben können. Mithilfe der neu berechneten Liste $KMEZZL(n)$ wird nun für den am besten geeigneten mobilen Endanwender eine erzwungene Delegation durchgeführt. Untervariante **DV2-2** der erzwungenen Delegationen hat sich aus Praxisanforderungen erge-

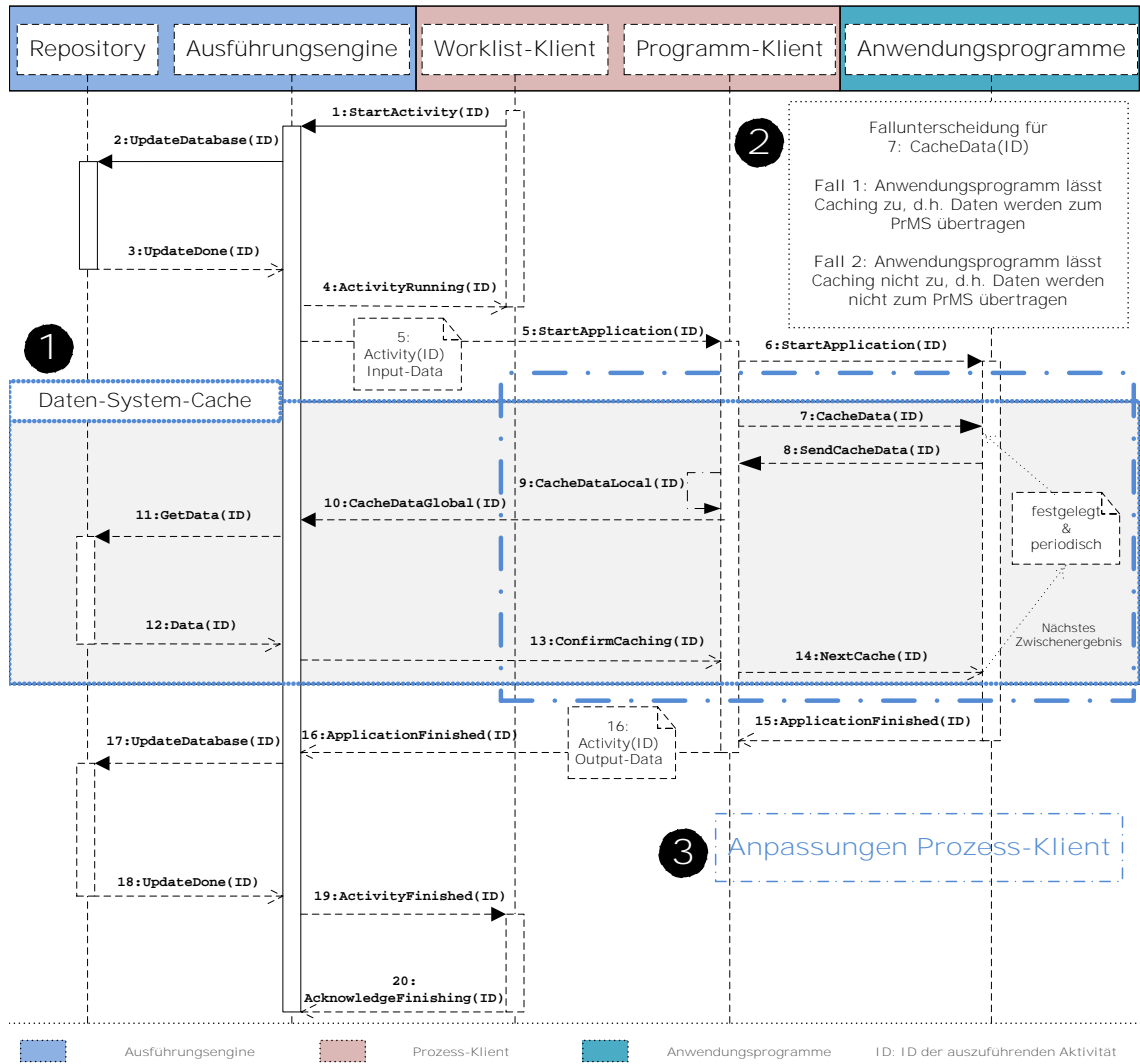


Abbildung 9.7: Ausführungsprotokoll und Zwischenspeicherung von Daten

ben (vgl. Kapitel 2). Für mobile Aktivitäten legt der Parameter MA_D einen Timer fest. Er besagt, dass innerhalb seiner festgelegten Zeitspanne die Ausführung der mobilen Aktivität abgeschlossen sein muss. Ist für MA_D ein Timer festgelegt, wird in zwei Fällen eine erzwungene Delegation durchgeführt.

1. Ein mobiler Endanwender hat eine mobile Aktivität angenommen bzw. gestartet und der Timer mit Zeitdauer MA_D ist abgelaufen; ferner gilt, dass sein Smart-Mobilgerät nicht durch das $PrMS$ kontaktiert werden kann.
2. Eine mobile Aktivität soll ausgeführt werden und wurde bisher von keinem mobilen Endanwender angenommen; ferner gilt, dass der Timer mit Zeitdauer MA_D abgelaufen ist.

Abschließend sollen für eine erzwungene Delegation die notwendigen Zustandsübergänge betrachtet werden. Eine erzwungene Delegation wird durch die Zustandsübergänge T_3 und T_{20}

Phase	Beschreibung	PK	PrMS
7 u. 8	Der Programm-Klient fordert beim ausführenden Anwendungsprogramm ein Zwischenergebnis an und bekommt dieses übermittelt.	✓	X
9	Der Programm-Klient speichert das vom Anwendungsprogramm übermittelte Ergebnis lokal ab.	✓	X
10	Der Programm-Klient sendet das Zwischenergebnis an das <i>PrMS</i> .	✓	X
11 – 13	Das <i>PrMS</i> nimmt das Zwischenergebnis an und bestätigt den Erhalt gegenüber dem Programm-Klienten.	X	✓
14	Der Programm-Klient startet die Speicherung des nächsten Zwischenergebnisses.	✓	X

X =trifft nicht zu, \checkmark =trifft zu
PK=Prozess-Klient, *PrMS*=Prozess-Management-System

Tabelle 9.6: Phasen des Ausführungsprotokolls bei Zwischenspeicherung von Daten

	Beschreibung	Auslöser
DV1	Delegation	Nichterreichbarkeit des Smart-Mobilgeräts.
DV2-1	Erzwungene Delegation	Maximale Anzahl möglicher Ablehnungen erreicht.
DV2-2	Erzwungene Delegation	Timer ist abgelaufen.

Tabelle 9.7: Delegationsvarianten

realisiert (vgl. Abb. 9.8). Durch T_3 wird die mobile Aktivität in den Zustand *Selected* versetzt und durch T_{20} direkt in den Zustand *Delegated*.

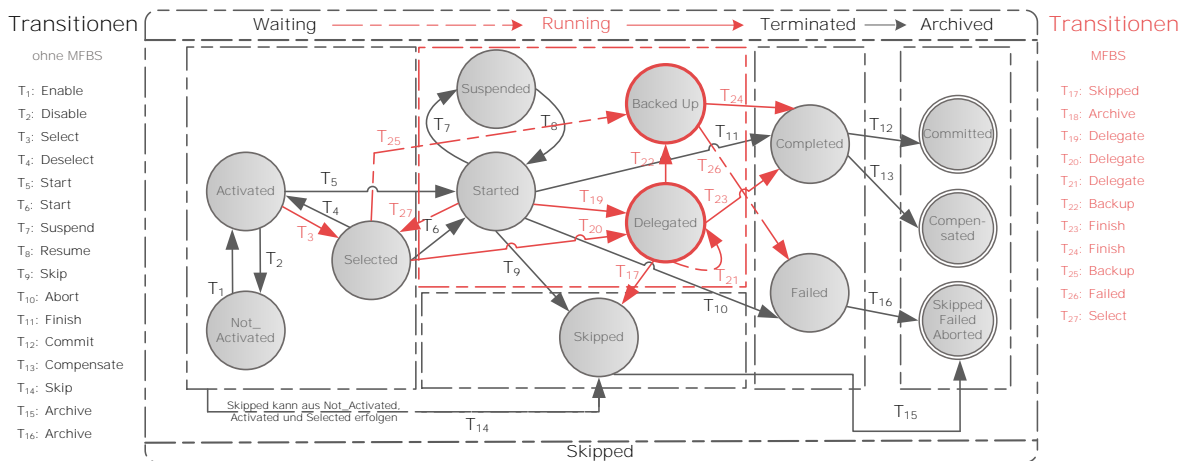


Abbildung 9.8: Erweitertes Zustandsmodell (vereinfacht)

9.5 Abbrechen mobiler Aktivitäten

Die zweite Methode zur Ausnahmebehandlung besteht entweder aus dem Abbrechen oder dem Backup der die Ausnahme verursachenden mobilen Aktivität. Diese Methode wird angewandt, wenn keine Delegation mehr möglich ist. Ferner wird dadurch die ursprüngliche Prozessausführung verändert. Was dies genau bedeutet und welche Voraussetzungen zu erfüllen sind, wird nachfolgend vorgestellt.

Zunächst muss definiert sein, wann keine Delegation mehr möglich sein soll. Dies ist genau dann der Fall, wenn durch Algorithmus 9.1 kein geeigneter mobiler Endanwender mehr für die Delegation gefunden werden kann. Für diesen Fall stellt sich die Frage, was die Ausnahmebehandlung unternehmen soll. In diesem Kontext wird *dtimeout* (Delegation Timeout) als weiterer Timer eingeführt: Kann Algorithmus 9.1 keinen geeigneten mobilen Endanwender mehr für eine Delegation finden, wird *dtimeout* gestartet. Wenn der Timer abgelaufen ist, versucht Algorithmus 9.1 erneut geeignete mobile Endanwender zu bestimmen. Gelingt dies wiederum nicht, startet schließlich Methode 2 der Ausnahmebehandlung (vgl. Abb. 9.1).

Wie entscheidet die Ausnahmebehandlung nun, ob eine mobile Aktivität abgebrochen werden oder ein Backup erstellt werden soll. Dazu wird geprüft, ob es sich bei der mobilen Aktivität, für die nun keine Delegation mehr möglich ist, um eine optionale oder zwingende (d.h. kritische) Aktivität handelt. Letzteres bedeutet, dass die mobile Aktivität aus Prozesssicht nicht abgebrochen werden darf, etwa wenn sie Prozessdaten erzeugt, die von nachfolgenden Aktivitäten gelesen werden. Würde man eine solche mobile Aktivität abbrechen, käme es in der Folge zu einer Ausnahme, die nicht mehr automatisch behoben werden kann, was aber im Sinne einer robusten Ausführung vermieden werden. Der mobile Kontext realisiert diese Unterscheidung durch den Parameter GP_{DA} . Ist dieser gesetzt, handelt es sich um eine kritische Aktivität und es wird ein Backup angewandt. Ist der Parameter GP_{DA} dagegen nicht gesetzt, ist die mobile Aktivität optional, d.h. sie kann auch problemlos abgebrochen werden. Wie in Abbildung 9.2 erkennbar, kann eine mobile Aktivität vom Zustand *Delegated* in den Zustand *Skipped* übergehen. Optionale Aktivitäten kommen in der Praxis häufig vor. In einer Klinik kann zum Beispiel eine mobile Aktivität ausdrücken, dass eine Schwester dokumentiert, ob der Patient sein Mittagessen bekommen hat. Wird dieser Eintrag durch den Abbruch der mobilen Aktivität nicht gesetzt, hat dies keine weiteren Auswirkungen.

Abschließend soll ein weiterer Aspekt des Parameters GP_{DA} diskutiert werden. Wird dieser Parameter automatisch ermittelt oder muss er manuell festgelegt werden? Die Praxissicht erfordert eine Kombination der Festlegungsart. Im Standardfall wird GP_{DA} automatisch, d.h. algorithmisch ermittelt. Die dazu notwendigen Algorithmen werden hier nicht vorgestellt (siehe [Rei00] für Details). Allerdings kann auch eine manuelle Festlegung notwendig werden. In der Praxis hat sich gezeigt, dass auch optionale Aktivitäten nicht abgebrochen werden dürfen. Wieder dient ein Beispiel aus der Klinik zur Erläuterung: Repräsentiert eine mobile Aktivität das Vorbereitungsgespräch für eine Operation, darf diese mobile Aktivität nicht abgebrochen werden, selbst wenn sie keine Daten schreibt. Im letztgenannten Fall würde die automatische Ermittlung ergeben, dass die mobile Aktivität abgebrochen werden kann. Um diesen Fall zu adressieren, kann bei der Modellierung mobiler Aktivitäten GP_{DA} explizit gesetzt werden. Wird GP_{DA} auch bei optionalen Aktivitäten gesetzt, wird für diese Aktivitäten ebenfalls ein Backup angewandt.

Abschließend sei erwähnt, dass es nicht möglich ist, bei der Modellierung mobiler Aktivitäten,

die Daten schreiben, GP_{DA} explizit abzuwählen. Die kombinierte Festlegungsart aus automatisch und manuell für den Parameter GP_{DA} hat sich in der Praxis als sehr effektiv erwiesen.

9.6 Backup mobiler Aktivitäten

Die Anwendung des Backups gehört ebenfalls zur zweiten Methode der Ausnahmebehandlung. Das Backup ergänzt das Abbrechen mobiler Aktivitäten. Es wird angewandt, wenn der Parameter GP_{DA} gesetzt ist und die Ausnahme bei einer zwingenden (d.h. kritischen) Aktivität auftritt. Analog zum Abbrechen mobiler Aktivitäten muss für das Backup $dtimeout$ abgelaufen sein (vgl. Abschnitt 9.5). Ein Backup stellt sicher, dass die Ausnahme einer kritischen mobilen Aktivität so behoben wird, als ob es sich um keine mobile Aktivität mehr handelt. Der mobile Kontext wird aufgehoben und die Ausführung auf ein stationäres Desktop-System übertragen. In der Folge kann die Ausnahme, die sich durch den mobilen Kontext ergibt, nicht wieder aus denselben (mobilen) Gründen auftreten. Tabelle 9.8 fasst wichtige Fragestellungen in Verbindung mit dem Backup zusammen.

	Backupaspekte	Erfordert
BA1	Was ist der Grundgedanke des Backups?	
BA2	Welche Backupvarianten sind aus praktischer Sicht zu unterscheiden?	
BA3	Welche weiteren Aspekte sind für den mobilen Endanwender bei der Durchführung des Backups zu berücksichtigen?	Betrachtung des Prozess-Klienten.
BA4	Welche weiteren Aspekte sind bezogen auf das $PrMS$ bei der Durchführung des Backups zu berücksichtigen?	Betrachtung des Ausführungsprotokolls.

Tabelle 9.8: Backupaspekte mobiler Aktivitäten

9.6.1 Fragestellung BA1: Grundgedanke

Ein Backup soll für eine kritische mobile Aktivität sicherstellen, dass diese in jedem Fall ausgeführt wird. Andernfalls können Folgeausnahmen auftreten, etwa dass notwendige Daten fehlen, die nicht automatisch behoben werden können. Weiters sei daran erinnert, dass ein Backup die Folge einer nicht mehr durchführbaren Delegation ist. Daher ist die Wahrscheinlichkeit hoch, dass aktuell kein mobiler Endanwender mehr zur Verfügung steht. Daher ist die Grundidee eines Backups, die Ausführung der mobilen Aktivität auf ein Desktop-System zu übertragen, sodass diese Ausführung sichergestellt ist. Dabei ist es essentiell, dass die Überführung automatisch erfolgt, sodass die robuste Ausführung mobiler Aktivitäten erhalten bleibt.

9.6.2 Fragestellung BA2: Backupvarianten

Ein Backup entspricht der Anwendung von Methode 2 (vgl. Abschnitt 9.2), d.h. die ursprüngliche Ausführung des Prozesses wird verändert. Nachfolgend werden die Änderungen für zwei Backupvarianten **BV1** und **BV2** vorgestellt.

	Beschreibung	Auslöser
BV1	Backup	Fehlgeschlagene Delegation einer mobilen Aktivität mit gesetztem Parameter GP_{DA} .
BV2	Beschleunigtes Backup	Fehlgeschlagene Delegation einer mobilen Aktivität mit gesetzten Parametern GP_{DA} und MA_D .

Tabelle 9.9: Backupvarianten

Das Konzept von Backupvariante BV1 zeigt Abbildung 9.9. Das *PrMS* fügt eine neue Aktivität **b1** und optional Aktivität **b2** zum Prozessablauf hinzu. Ist die Delegation bei Aktivität **b** fehlgeschlagen, wird diese abgebrochen und der Prozessablauf geht automatisch mit Aktivität **b1** weiter. Im Gegensatz zu **b**, wird **b1** jedoch auf einem Desktop-System durchgeführt. Ferner ist **b1** so definiert, dass diese Aktivität die gleichen Daten wie die mobile Aktivität **b** erzeugt, sodass diese später von Aktivität **c** gelesen werden können. Die Endanwender, die zur Ausführung von **b1** infrage kommen, sind alle für die Ausführung dieser Aktivität berechtigten Endanwender. Die Rechte "erbt" diese Aktivität von der ursprünglichen mobilen Aktivität **b**. Ferner wird eine optionale Aktivität **b2** hinzugefügt, wenn die Backupvariante BV1 bei der Modellierung entsprechend konfiguriert wurde. Dies wird über einen neuen Kontextparameter MA_V geregelt, der funktional vom Parameter GP_{DA} abhängig ist. MA_V muss bei der Modellierung manuell festgelegt werden. Ist MA_V für eine mobile Aktivität gesetzt, wird die Aktivität **b2** nach Aktivität **b1** ausgeführt. Aufgabe von **b2** ist es, das Ergebnis von Aktivität **b1** zu validieren. Validation heißt, dass der mobile Endanwender, bei dem die Delegation fehlschlug, die Ausführung von Aktivität **b1** validieren muss. Dazu werden ihm die von **b1** erzeugten Ergebnisse angezeigt. Er muss dann die Entscheidung treffen, ob die Daten valide sind. Dieser Schritt dient dazu, die mobile Ausführung und die Ausführung auf dem Desktop-System manuell abzugleichen. Das Ergebnis wird in einem Datenelement vermerkt (vgl. Abb. 9.9). Wichtig ist, dass es nur vermerkt wird und nicht zu einer erneuten Ausführung von **b1** führt. D.h. es soll lediglich dokumentiert werden, dass die beiden Ausführungen nicht harmonisieren. Wie mit einer nicht erfolgreichen Validation umgegangen wird, muss manuell festgelegt werden. Abschließend sei erwähnt, dass die Validation sowohl mobil als auch stationär erfolgen kann. Das *PrMS* versucht dazu den Endanwender, der für die fehlgeschlagene Delegation verantwortlich ist, auf beiden Wegen zu kontaktieren.

Abbildung 9.10 stellt Backupvariante BV2 vor. Sie kommt zur Anwendung, wenn die Delegation fehlschlug und ferner der Parameter MA_D für die mobile Aktivität mit einem Wert versehen ist. Im Gegensatz zur Backupvariante BV1 werden die beiden Aktivitäten **b1** und **b2** parallel zur der mobilen Aktivität eingefügt, welche die Delegationsausnahme verursacht. Dies wirft die Frage auf, warum die Einfügung hier parallel erfolgt, zumal erst nach der fehlgeschlagenen Delegation entschieden werden kann, ob ein Backup notwendig ist. Generell ist dies richtig, jedoch stellt Backupvariante BV2 zusätzlich sicher, dass bei einer gesetzten Dringlichkeit auch das Backup-Prozedere beschleunigt wird. Aus diesem Grund wendet Backupvariante BV2 ein fortschrittliches Verfahren an. Bereits beim Start der mobilen Aktivität **b** startet das *PrMS* auch Aktivität **b1**. Dazu werden alle infrage kommenden Endanwender ermittelt und die Aktivität wird in deren Worklists auf den Desktop-Systemen eingestellt. Jedoch sehen die Desktop-Anwender nur, dass sich die Aktivität in Ausführung befindet, annehmen können sie diese allerdings nicht. Diese Maßnahme bewirkt zwei Dinge. Erstens sind die möglichen En-

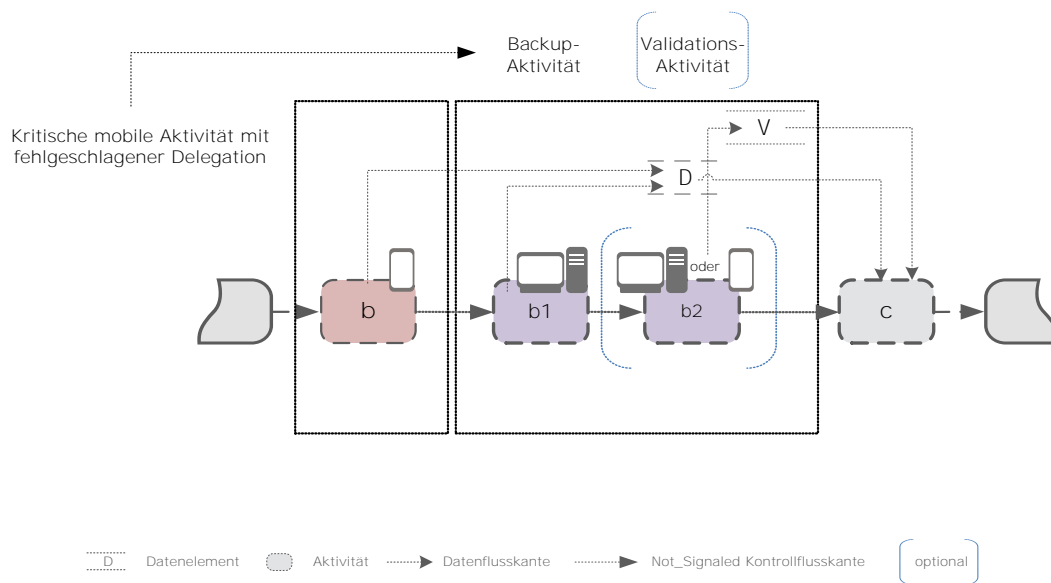


Abbildung 9.9: Backupvariante BV1: Backup

danwender für das Backup bereits bei Eintreten des Backups berechnet. Zweitens sehen die infrage kommenden Endanwender auf den Desktop-Systemen die Statusänderung sofort, wenn die mobile Aktivität beendet bzw. eine Delegation durchgeführt wird. Schlägt die Delegation fehl, wird bei den stationären Endanwendern der Status verändert, sodass diese **b1** ausführen können. Wird die mobile Aktivität **b** ordnungsgemäß beendet, wird die Aktivität **b1** aus den Worklists der stationären Endanwender wieder entfernt.

Bei Backupvariante BV2 kann die Zuteilung der Aktivität **b1** schneller erfolgen als bei Backupvariante BV1. Zusätzlich sehen die stationären Endanwender die Zustandsänderung. Dieser Umstand hat sich in Praxistests bewährt. Die Validationsaktivität wird bei Backupvariante BV2 gleichermaßen wie bei Backupvariante BV1 durchgeführt.

Abschließend soll erwähnt werden, dass Backupvariante BV2 umfangreiche Berechnungen erfordert, die eventuell auch bei einem gesetzten Parameter MA_D nicht gewollt sind. Daher muss manuell entscheiden werden können, dass anstelle von Backupvariante BV2 die Backupvariante BV1 durchgeführt wird. Außerdem kann Backupvariante BV2 auch für mobile Aktivitäten sinnvoll sein, die keine Dringlichkeit vorweisen. In diesem Fall muss MA_D manuell bei der Modellierung gesetzt werden.

9.6.3 Fragestellung BA3: Wichtige Aspekte für den Endanwender

Bei der praktischen Durchführung eines Backups ist ein Aspekt wichtig: Wird das Backup erforderlich, muss dieses dem die Ausnahme verursachenden mobilen Endanwender, bei dem die Delegation nicht durchgeführt werden konnte, entsprechend mitgeteilt werden. Hat dieser mobile Endanwender wieder eine Verbindung zum *PrMS*, wird der Status jener Aktivitäten, die nun im Zustand des Backups sind, in der Worklist entsprechend aktualisiert (vgl. Abb. 9.11①②). Dies erfolgt analog zum Aktualisieren der Worklist bei einer Delegation (vgl. Abschnitt 9.4.3).

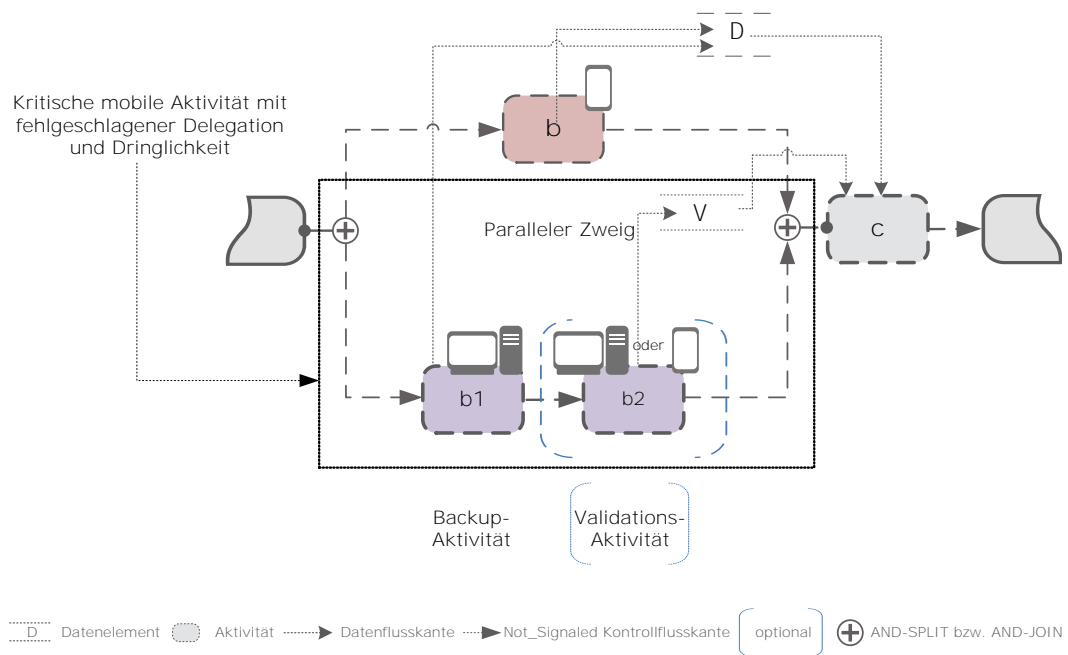


Abbildung 9.10: Backupvariante BV2: Beschleunigtes Backup

Zwei konkrete Zustände sind für beide Backupvarianten möglich. Erstens kann sich eine Aktivität im Zustand *In-Backup* befinden, d.h. der mobile Endanwender kann die Aktivität wieder zurückfordern. Zweitens kann die Aktivität im Zustand *Backed Up* sein. Dann wird sie durch das Backup ausgeführt und der mobile Endanwender hat keinerlei Einflussmöglichkeit mehr. Allerdings muss er im zweitgenannten Fall ggf. noch die Validationsaktivität ausführen. Für diesen Fall wird ihm die Aktivität daher mittels einer erzwungenen Delegation in die Worklist gestellt.

Bei der Durchführung des Backups wird der Endanwender auf dem Desktop-System anders ermittelt als bei einer Delegation. Beim Backup wird nicht nur ein Endanwender ausgewählt, der dann das Backup annehmen oder ablehnen kann, sondern alle möglichen stationären Endanwender bekommen die Aktivität in ihre Worklist anstehender Aktivitäten auf ihrem Desktop-System eingestellt. Da ab diesem Zeitpunkt kein Kontext mehr zu berücksichtigen ist, kommen prinzipiell alle Endanwender, wie bei traditionellen *PrMS* üblich, gleichermaßen infrage.

9.6.4 Fragestellung BA4: Wichtige Aspekte für das PrMS

Für das Backup wird eine Transformation des ursprünglichen Prozessmodells nötig. Diese wurde in den Abbildungen 9.9 und 9.10 vorgestellt. Generell sind dynamische Strukturänderungen mit erheblichem Aufwand verbunden [WRRM08]. Die strukturellen Änderungen der beiden Backupvarianten sind jedoch immer dieselbe, zudem müssen keine strukturellen Konflikte betrachtet werden. Daher kann die skizzierte Prozessmodelltransformation zur Ausnahmebehandlung als vertretbar angesehen werden.

Ein weiterer Aspekt betrifft die Listenberechnungen zur Auswahl geeigneter Endanwender bei den Backups. Abbildung 9.12 zeigt die Listenberechnungen für Backupvariante BV1. Für Ak-

Phase 2: Mobiler Endanwender ist wieder mit PrMS verbunden

Aktivität	Status	Zeitlimit	Rückfordern?
Röntgenbilder Frau Maier anschauen	In-Backup	1 4h 30min	<input type="radio"/> Ja <input type="radio"/> Nein
Bestätige Erhalt Blutuntersuchung Horsch	Backed Up	2 13:30 Uhr	<input type="radio"/> Ja <input type="radio"/> Nein
Befundung Frau Zemas	Backed Up	2 -----	<input type="radio"/> Ja <input type="radio"/> Nein
Laborwerte Herr Zweller anfordern	In-Backup	1 0h 0min	<input type="radio"/> Ja <input type="radio"/> Nein

Marc Schickler:
Backup-Status

Abbildung 9.11: Prozess-Klient und Backup-Zustand

tivität **b** kommt Algorithmus 9.1 zum Einsatz (vgl. Abschnitt 9.4.2) und für Aktivität **b1** Algorithmus 5.1. Die von Algorithmus 5.1 ermittelten Endanwender, die am *PrMS* durch ein Desktop-System angemeldet sind, bekommen schließlich Aktivität **b1** in ihre Worklists eingestellt. Für Aktivität **b2** sind keine Berechnungen notwendig, da es sich um den mobilen Endanwender handelt, bei dem die Delegation fehlschlug.

Abbildung 9.13 zeigt die Listenberechnungen für Backupvariante BV2. Die Berechnungsgrundlagen für Backupvariante BV2 unterscheiden sich nicht von denen von Backupvariante BV1. Jedoch wird bei Backupvariante BV2 die Listenberechnung für Aktivität **b1** bereits im Zustand *Selected* durchgeführt. Ferner wird die Aktivität bis zum Zustand *Backed Up* in den Worklists der Endanwender gesperrt⁶

9.7 Weitere Aspekte

Dieser Abschnitt diskutiert weitere Aspekte der Behandlung von Ausnahmen in Verbindung mit mobilen Aktivitäten.

⁶Durch das Schloss symbolisiert.

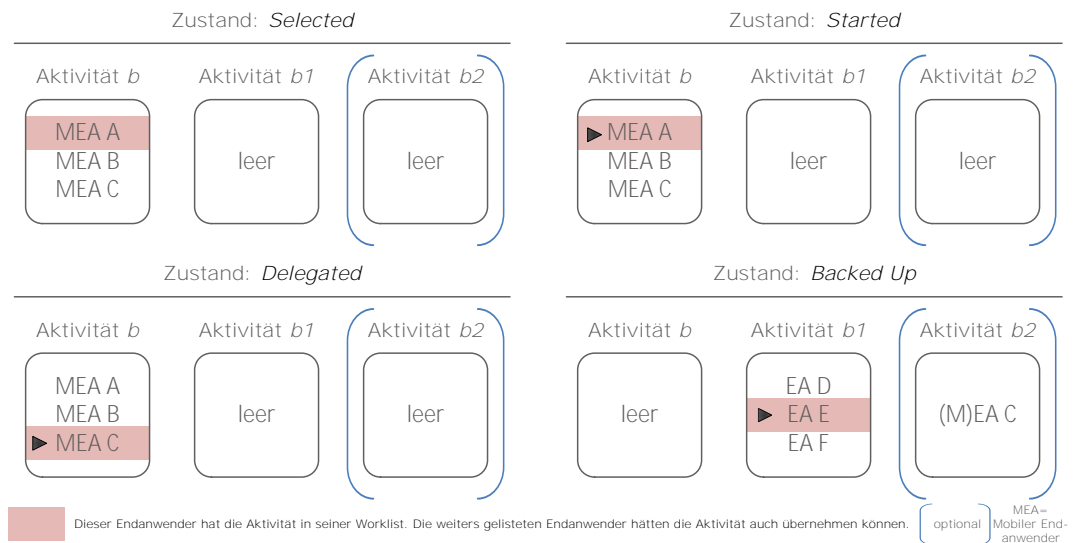


Abbildung 9.12: Notwendige Listenberechnungen für Endanwender der Backupvariante BV1

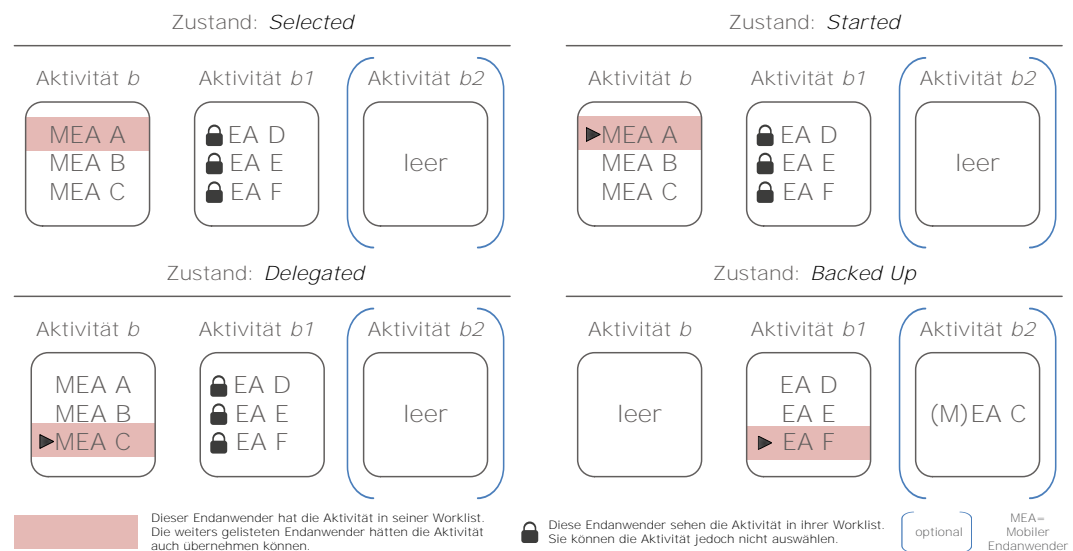


Abbildung 9.13: Notwendige Listenberechnungen für Endanwender der Backupvariante BV2

9.7.1 Offline-Aktivitäten

Wie in Kapitel 8 gezeigt, zeigt ein gesetzter Parameter MA_{OFF} an, dass die mobile Aktivität Offline ausgeführt werden kann. Die Offline-Ausführung muss auch im Kontext von Ausnahmebehandlungen berücksichtigt werden. Ferner wurde gezeigt, dass MA_{OFF} funktional vom Parameter MA_D abhängt. Solange der Timer, welcher durch MA_D bestimmt wird, nicht abgelaufen ist, kann die mobile Aktivität bei gesetztem Parameter MA_{OFF} entsprechend Offline ausgeführt werden, ohne dass eine Ausnahmebehandlung angestoßen wird.

Was geschieht nun, wenn die durch MA_D festgelegte Dauer ausläuft? Schließlich soll verhindert werden, dass die Offline-Situation unendlich lange dauert. Dazu wird folgendes Prozedere

durchlaufen: Nach Verstreichen der von MA_D festgelegten Dauer, kontaktiert das $PrMS$ den jeweiligen mobilen Endanwender, der die Offline-Aktivität ausführt. In der Folge werden zwei Fälle unterschieden. Im Fall 1 ist der mobile Endanwender erreichbar. Hier wird ihm nach Verstreichen der von MA_D festgelegten Dauer vorgeschlagen, die Offline-Aktivität in Kombination mit einem Systemwechsel (vgl. Kapitel 8) durchzuführen. Stimmt er zu, wird keine Ausnahmebehandlung durchgeführt und es werden keine weiteren Schritte unternommen. Stimmt er nicht zu, wird die Aktivität ab diesem Zeitpunkt als normale mobile Aktivität betrachtet und der Timer *atimeout* gestartet. Auf Grundlage dieses Timers wird die reguläre Ausnahmebehandlung gestartet, die bei Auslaufen von *atimeout* initiiert wird (vgl. Abschnitt 9.4.1). Im Fall 2 ist der mobile Endanwender nicht erreichbar. Dann wird ebenfalls die reguläre Ausnahmebehandlung gestartet, als wäre *atimeout* abgelaufen (vgl. Abschnitt 9.4.1).

9.7.2 Parametergewichtung bei Delegationen

Zuletzt soll ein Aspekt bezogen auf die Delegation diskutiert werden. In Abschnitt 9.4.2 wurde Algorithmus 9.1 vorgestellt, der die Liste geeigneter mobiler Endanwender für eine Delegation berechnet. Die Listeneinträge haben einen Priorisierungswert, der in den Zeilen 6, 10 und 14 durch Algorithmus 9.1 berechnet wird. Wie erkennbar, besteht die Berechnung aus 3 Komponenten, die den Priorisierungswert schließlich bestimmen. Die erste Komponente berücksichtigt das Delegationsverhalten (ME_{DB}), die zweite das Ressourcenverhalten (ME_{RB}) und die dritte den Instant-Shutdownzähler ME_{ISZ} . Eine weitere Optimierung für die Praxis könnte darin bestehen, die Komponenten für spezifische Szenarien zu gewichten. So könnte in einem bestimmten Szenario der Instant-Shutdownzähler gar nicht gewertet werden oder eventuell das Ressourcenverhalten mehr Gewicht bekommen.

9.7.3 Übergabe von Aktivitäten

Mithilfe des Delegationsprinzips ist es möglich, eine Übergabe mobiler Aktivitäten von einem Endanwender zum nächsten zu realisieren. Dazu kann er in seiner Worklist die Funktion Delegation auswählen, das System sucht dann einen geeigneten mobilen Endanwender für die Übergabe aus. Für die Auswahl kommen die gleichen Kontextregeln und derselbe Algorithmus wie bei einer Delegation zum Einsatz. Hat ein mobiler Endanwender einen bestimmten mobilen Endanwender im Auge, muss er diesen explizit benennen. Die verwendeten Konzepte der Delegation, in diesem Fall das angewandte Protokoll, unterstützen auch diesen Anwendungsfall.

9.7.4 Ausführungsprotokoll und Parameterübertragung

In diesem Kapitel wurden weitere Parameter eingeführt (z.B. *atimeout*), die auch beim Ausführungsprotokoll berücksichtigt werden müssen. Diese werden zwischen dem $PrMS$ und dem Smart-Mobilgerät ausgetauscht. Abbildung 9.14 zeigt die Stellen des Protokolls, an denen ein Austausch stattfindet. Tabelle 9.10 führt die entsprechenden Parameter bezogen auf die markierten Stellen auf. Auf Basis dieser Übertragungen hat das $PrMS$ stets einen möglichst aktuellen Status des Smart-Mobilgeräts. Ferner legen die Parameter MA_{SFOS} und MA_{AZOS} fest, wie oft die Schritte 6-9 durchgeführt werden.

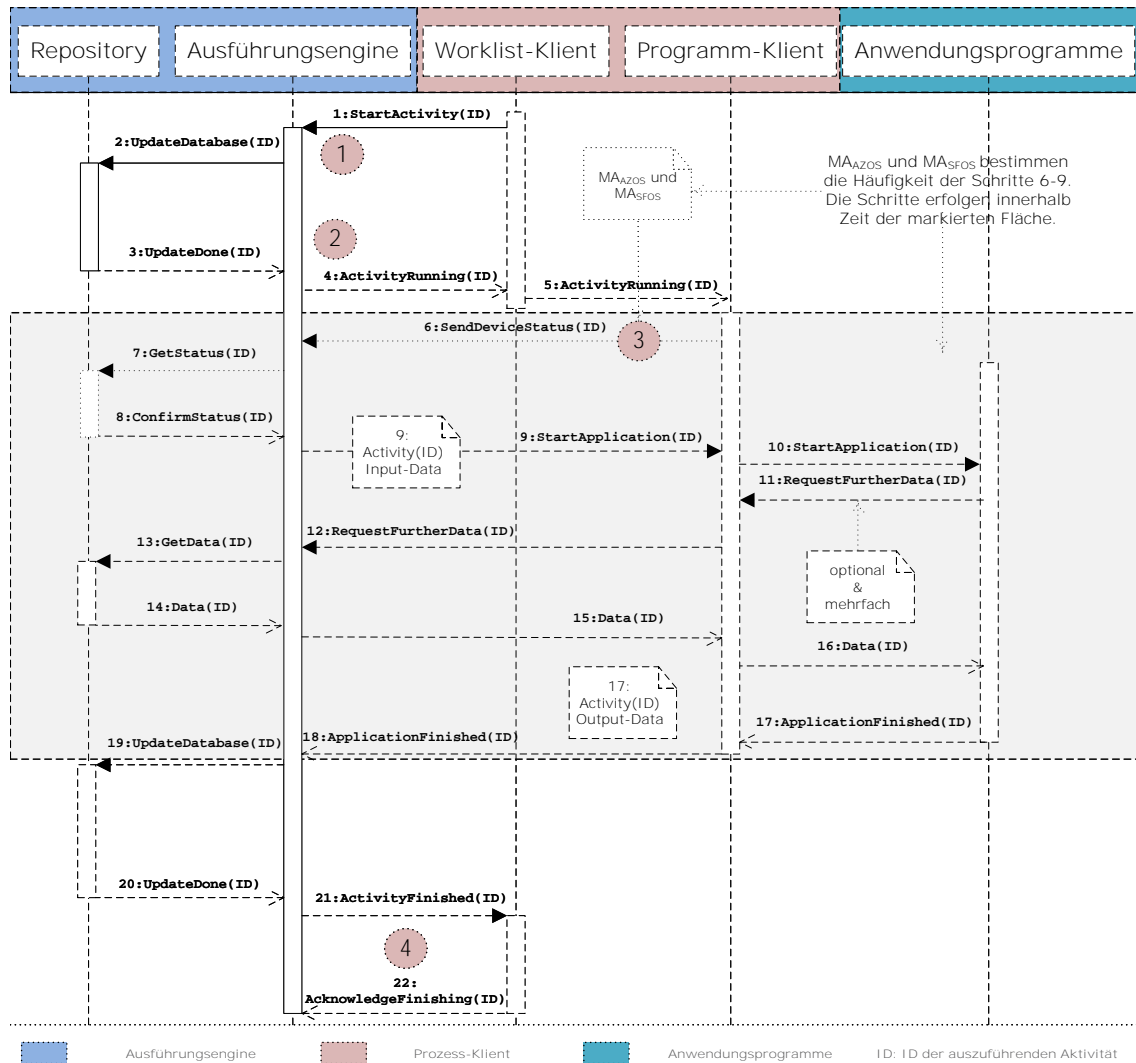


Abbildung 9.14: Parameterübertragung

9.7.5 Ausführungshistorie

Die Ausnahmebehandlung wird auch mit neuen Zustandsmarkierungen für sich in Ausführung befindliche bzw. beendete mobile Aktivitäten vermerkt (vgl. Abb. 9.15). Es kommt eine Zustandsmarkierung für eine durchgeführte Delegation hinzu und eine weitere für das Backup. Die Zustandsmarkierung für das Abbrechen mobiler Aktivitäten ist bereits durch den Zustand *Skipped* abgedeckt (vgl. Kapitel 5). In Abbildung 9.15 wird sowohl die Zustandsmarkierung für das Abbrechen als auch die Delegation und das Backup mobiler Aktivitäten farblich markiert.

Die Ausnahmebehandlung wird ebenfalls in der Ausführungshistorie (vgl. Kapitel 5) vermerkt. Eine vereinfachte Ausführungshistorie für die Prozessinstanz aus Abbildung 9.15 sieht wie folgt aus:

9.8 Diskussion

Dieser Abschnitt diskutiert die Ausnahmebehandlung zur robusten Ausführung mobiler Aktivitäten. Weiters werden verwandte Ansätze diskutiert.

9.8.1 Beitrag zur robusten Ausführung mobiler Aktivitäten

Die Ausnahmebehandlung konnte viele Anforderungen dieser Arbeit erfüllen (vgl. Kapitel 2). Tabelle 9.11 fasst diese zusammen und zeigt, welche Maßnahmen zur Erfüllung der jeweiligen Anforderung beitragen. Ferner wurden neue Kontextparameter für die Ausnahmebehandlung eingeführt. Die Kontextparameter ME_{DPV} , ME_{DB} und ME_{RB} beziehen sich auf den mobilen Endanwender und berücksichtigen spezifisch dessen Verhalten für die Ausnahmebehandlung. Der Kontextparameter MA_V wiederum bezieht sich auf die mobile Aktivität und realisiert die Durchführung der Validationsaktivität.

Anforderung	Maßnahmen
Funktionale Anforderungen mobiler Aktivitäten	
Anforderung 2	Die Ausnahmebehandlung berücksichtigt den mobilen Kontext. Dazu auch Einführung des neuen Parameters MA_V .
Anforderung 5	Die Übergabe von Aktivitäten wird durch das Delegationsprinzip ermöglicht (vgl. Abschnitt 9.7.3).
Anforderung 9	Die Ausnahmebehandlung berücksichtigt Offline-Aktivitäten (vgl. Abschnitt 9.7.1).
Funktionale Anforderungen des mobilen Kontexts	
Vergleiche zu den Anforderungen 10-15 die Tabelle 9.3 und den Algorithmus 9.1	
Anforderung 10	Die Ausnahmebehandlung berücksichtigt den Ausführungsort.
Anforderung 11	Die Ausnahmebehandlung berücksichtigt den Netztyp.
Anforderung 12	Die Ausnahmebehandlung berücksichtigt den Formfaktor.
Anforderung 13	Die Ausnahmebehandlung berücksichtigt den Energiestatus.
Anforderung 14	Die Ausnahmebehandlung berücksichtigt die Dringlichkeit.
Anforderung 15	Die Ausnahmebehandlung berücksichtigt das Endanwenderverhalten. Dazu auch Einführung der neuen Parameter ME_{DPV} , ME_{DB} und ME_{RB} .
Nicht-funktionale Anforderungen mobiler Aktivitäten	
Anforderung 16	Die angewandte Methodik (vgl. Abschnitt 9.2) der <i>MABS</i> ermöglicht die robuste Ausführung mobiler Aktivitäten.
Anforderung 17	Die Ausnahmebehandlung erfordert bis auf die Annahme oder Rückforderung einer Delegation bzw. eines Backups keine manuellen Entscheidungen.

Tabelle 9.11: Beitrag der Ausnahmebehandlung zur robusten Ausführung mobiler Aktivitäten

9.8.2 Verwandte Ansätze

Nur wenige Ansätze verfolgen eine explizite Ausnahmebehandlung in Verbindung mit mobilen Aktivitäten. In [ZHKL10] wird die Notwendigkeit einer Ausnahmebehandlung ins Metamodell für mobile Aktivitäten aufgenommen, ohne jedoch konkrete Strategien vorzustellen. Dennoch wird die Wichtigkeit betont. In [HS09] wird eine explizite Ausnahmebehandlung für mobile Aktivitäten vorgestellt. Jedoch werden ausschließlich automatische Aktivitäten, nicht aber interaktive Aktivitäten behandelt. Ferner geht [HS09] davon aus, dass nur ein mobiler Endanwender am Prozess beteiligt ist.

9.9 Zusammenfassung

In diesem Kapitel wurde die Ausnahmebehandlungsstrategie mobiler Aktivitäten vorgestellt. Zuerst wurde vorgestellt, welche Ausnahmen adressiert werden. Weiters wurde gezeigt, wie die Ausnahmebehandlung sicherstellt, dass die Behandlung der Ausnahme die ursprüngliche Ausführung beibehält. Scheitert selbst diese Maßnahme, wurde gezeigt, wie die Ausnahmebehandlung durch Prozesstransformationen oder Abbrechen mobiler Aktivitäten komplexe Ausnahmeszenarios berücksichtigt. Insbesondere fortschrittlich an der vorgestellten Ausnahmebehandlung sind die nahezu vollautomatische Durchführung und die dedizierte Berücksichtigung des mobilen Kontexts.

Wer Freiheit opfert, um Sicherheit zu gewinnen, wird beides verlieren.

Benjamin Franklin (1706-1790)

10

Mobile Aktivitäten und Prozess-Constraints

Dieses Kapitel betrachtet für die Praxis besonders relevante Prozess-Constraints im Kontext mobiler Aktivitäten. Ein Prozess-Constraint liegt zum Beispiel vor, wenn zwei mobile Aktivitäten von verschiedenen mobilen Endanwendern ausgeführt werden sollen (sog. Vier-Augen-Prinzip [VAP14]). Dieser Constraint wird zum Beispiel im klinischen Alltag häufig gefordert: Fällt ein Medikament unter das Betäubungsmittelgesetz [BTM14b], darf eine Schwester es nur verabreichen, wenn dieses von einem Arzt zuvor genehmigt worden ist [BTM14a].

Die Realisierung solcher Prozess-Constraints ist nicht trivial. Einerseits erfordert ihre Sicherstellung bei der Ausführung mobiler Aktivitäten und eventueller Ausnahmebehandlungen (vgl. Kapitel 9) besondere Vorkehrungen. Andererseits sind für die Ermittlung der potentiellen Endanwender im Kontext der Bearbeitung einer mobilen Aktivität komplexe Berechnungen durchzuführen.

Dieses Kapitel stellt drei typische Prozess-Constraints vor und diskutiert, wie sich diese geeignet mit den Konzepten mobiler Aktivitäten verknüpfen lassen. Existierende Ansätze zu Prozess-Constraints (z.B. [BFA97, BFA99]) klammern einen solchen mobilen Prozesskontext weitgehend aus.

Kapitel 10 gliedert sich wie folgt: In Abschnitt 10.1 werden besonders wichtige Prozess-Constraints eingeführt. Deren Praxisrelevanz diskutiert Abschnitt 10.2. Abschnitt 10.3 wiederum erörtert, was für die Erzeugung und Ausführung mobiler Aktivitäten zu berücksichtigen ist, wenn zusätzlich Prozess-Constraints adressiert werden. Abschnitt 10.4 befasst sich mit Prozess-Constraints im Kontext der in Kapitel 9 eingeführten Ausnahmebehandlung für mobile Aktivitäten. Abschnitt 10.5 behandelt weitere Aspekte und Abschnitt 10.6 diskutiert verwandte Ansätze. Abschnitt 10.7 fasst das Kapitel zusammen.

10.1 Einleitung

In Kapitel 5 wurden die berücksichtigten Prozess-Constraints im Kontext des verwendeten Metamodells bereits erwähnt (s. Tabelle 5.2 EA3-EA5). Dieses Kapitel behandelt diese drei Prozess-Constraints nun im Kontext mobiler Aktivitäten. Im Folgenden werden die Prozess-

Constraints zwecks besserer Lesbarkeit mit ihrem Namen (bzw. dessen Abkürzung) referenziert (vgl. Tabelle 10.1).

	Bezeichnung	Abkürzung	Bedeutung
EA3	Binding of Duties	BoD	Zwei Aktivitäten müssen vom selben Endanwender durchgeführt werden.
EA4	Separation of Duties	SoD	Zwei Aktivitäten müssen von verschiedenen Endanwendern durchgeführt werden.
EA5	Cardinality	Card	Eine Aktivität muss mehrfach hintereinander ausgeführt werden.

Tabelle 10.1: Prozess-Constraints für Endanwender (vereinfacht)

10.2 Praxisrelevanz

Zu den Prozess-Constraints aus Tabelle 10.1 wurden Praxisbeispiele in mehreren Anwendungsdomänen analysiert, um festzustellen, ob sie auch im Kontext mobiler Aktivitäten relevant sind. Tabelle 10.2 zeigt einen Ausschnitt dieser fachlichen Analysen der Prozess-Constraints *SoD*, *BoD* und *Card*. Des Weiteren wurden Praxisszenarien analysiert, in denen auch eine kombinierte Anwendung der Prozess-Constraints relevant ist. So kann zum Beispiel ein **BoD**- mit einem **SoD**-Constraint kombiniert werden, wie in Abbildung 10.1 für die Aktivitäten *a*, *c* und *d* exemplarisch gezeigt wird. Tabelle 10.3 illustriert ausgewählte Kombinationen von Prozess-Constraints. Ferner zeigt Abbildung 10.2, stellvertretend für alle Constraint-Kombinationen mit einem **Card**-Constraint, die Semantik solcher Kombinationen. Die in Tabelle 10.3 aufgeführten Constraint-Kombinationen sind die aus Praxissicht besonders relevanten. Weitere Kombinationen, wie zum Beispiel ein **BoD-BoD-BoD**-Constraint, werden in der Folge nicht weiter betrachtet (vgl. Abb. 10.3). Einerseits treten sie in der Praxis nur selten auf, andererseits können sie in der Mehrzahl der Fälle durch eine alternative Modellierung des Szenarios in anderer Art und Weise realisiert werden.

10.3 Anpassungen bei der Erzeugung und Ausführung mobiler Aktivitäten

Dieser Abschnitt behandelt Aspekte, die für Prozess-Constraints im Kontext der Erzeugung und Ausführung mobiler Aktivitäten berücksichtigt werden müssen. Für die Modellierung mobiler Aktivitäten müssen bei der Verwendung von Prozess-Constraints keine speziellen Vorkehrungen getroffen werden. Einerseits werden Prozess-Constraints bereits durch das verwendete Prozess-Metamodell berücksichtigt (vgl. Kapitel 5). Andererseits werden durch sie keine neuen Parameter für den mobilen Kontext eingeführt. Tabelle 10.4 gibt eine Übersicht, der bezogen auf die Erzeugung bzw. Ausführung mobiler Aktivitäten im Zusammenhang mit den Prozess-Constraints *SoD*, *BoD* und *Card* anzupassenden Konzepte. Weiters zeigt Tabelle 10.4, für welche Aktivität des Prozess-Constraints die Anpassung erfolgt. Die Prozess-Constraints *SoD*

Anwendungsdomäne	SoD	BoD	Card
Gesundheitswesen <i>Klinik</i> [PTKR10, PTR10, PLRH12, PMLR14]	Die Aktivitäten <i>Gutachten erstellen</i> und <i>Gutachten abnehmen</i> müssen verschiedene Ärzte durchführen.	Die Aktivitäten <i>EKG abnehmen</i> und <i>EKG auswerten</i> muss derselbe Arzt durchführen.	Die Aktivität <i>Blutdruck messen</i> muss mehrfach hintereinander ausgeführt werden.
Gesundheitswesen <i>Rettungsdienst</i> [KPR12b, KPR12a]	Die Aktivitäten <i>Morphium abholen</i> und <i>Morphium verabreichen</i> müssen verschiedene Sanitäter durchführen.	Die Aktivitäten <i>Patienten überwachen</i> und <i>Patienten übergeben</i> muss derselbe Sanitäter durchführen.	Die Aktivität <i>Sauerstoffsättigung messen</i> muss mehrfach ausgeführt werden.
Luftfahrt <i>Catering</i> [Sto11, Sto10]	Die Aktivitäten <i>Essensbehälter befüllen</i> und <i>Essensbehälter überprüfen</i> müssen verschiedene Sachbearbeiter durchführen.	Die Aktivitäten <i>Essensbehälter entgegennehmen</i> und <i>Essensbehälter reinigen</i> muss derselbe Sachbearbeiter durchführen.	Die Aktivität <i>Temperatur des Essensbehälters messen</i> muss mehrfach ausgeführt werden.
Kreditbearbeitung <i>Finanzwesen</i> [JS98, SG94, All08, MH08]	Die Aktivitäten <i>Kredit bearbeiten</i> und <i>Kredit bewilligen</i> müssen verschiedene Sachbearbeiter durchführen.	Die Aktivitäten <i>Kredit bearbeiten</i> und <i>Krediturkunde übergeben</i> muss derselbe Sachbearbeiter durchführen.	Die Aktivität <i>Kreditprospekte versenden</i> muss mehrfach ausgeführt werden.
Automobilindustrie <i>Fahrzeugentwicklung</i> [BBP09]	Die Aktivitäten <i>Bauteil ändern</i> und <i>Bauteiländerung genehmigen</i> müssen verschiedene Sachbearbeiter durchführen.	Die Aktivitäten <i>Bauteiländerung überwachen</i> und <i>Bauteiländerung dokumentieren</i> muss derselbe Sachbearbeiter durchführen.	Die Aktivität <i>Bauteiländerung überprüfen</i> muss mehrfach ausgeführt werden.
Klinische Studien <i>Studienanfertigung</i> [SRLP ⁺ 13, SSP ⁺ 13, RLPL ⁺ 13, IRLP ⁺ 13, CNB ⁺ 13, SSP ⁺ 14]	Die Aktivitäten <i>Englische Fragen bearbeiten</i> und <i>Deutsche Fragen bearbeiten</i> müssen von verschiedenen Psychologen bearbeitet werden.	Die Aktivitäten <i>Patienten befragen</i> und <i>Daten auswerten</i> muss derselbe Psychologe durchführen.	Die Aktivität <i>Gemütszustände bestimmen</i> muss mehrfach ausgeführt werden.

Tabelle 10.2: Praxisrelevanz von Prozess-Constraints

und *BoD* referenzieren jeweils zwei Aktivitäten (A1,A2)¹, wohingegen sich der **Card** Prozess-Constraint stets auf exakt eine Aktivität bezieht.

10.3.1 Anpassung 1: Zuteilungsschema BoD-Aktivität A1

Bei der Erzeugung mobiler Aktivitäten, deren Ausführung durch ein **BoD**-Constraint eingeschränkt wird, wird das Zuteilungsschema mobiler Endanwender zu den mobilen Prozessaktivitäten geändert. Genauer gesagt erfolgt die Anpassung des Zuteilungsschemas nur für die erste Aktivität des **BoD**-Constraints. Bei der zweiten Aktivität ist keine Entscheidung bzgl der Zuteilung mehr zu treffen, da infolge des **BoD**-Constraints durch die erste Aktivität auch der

¹Das heißt, A1 ist die *erste* und A2 die *zweite* Aktivität des Prozess-Constraints. Im Laufe dieses Kapitels wird per Konvention mit A1 (A2) die erste (zweite) Aktivität eines *SoD*- oder *BoD*-Constraints bezeichnet.

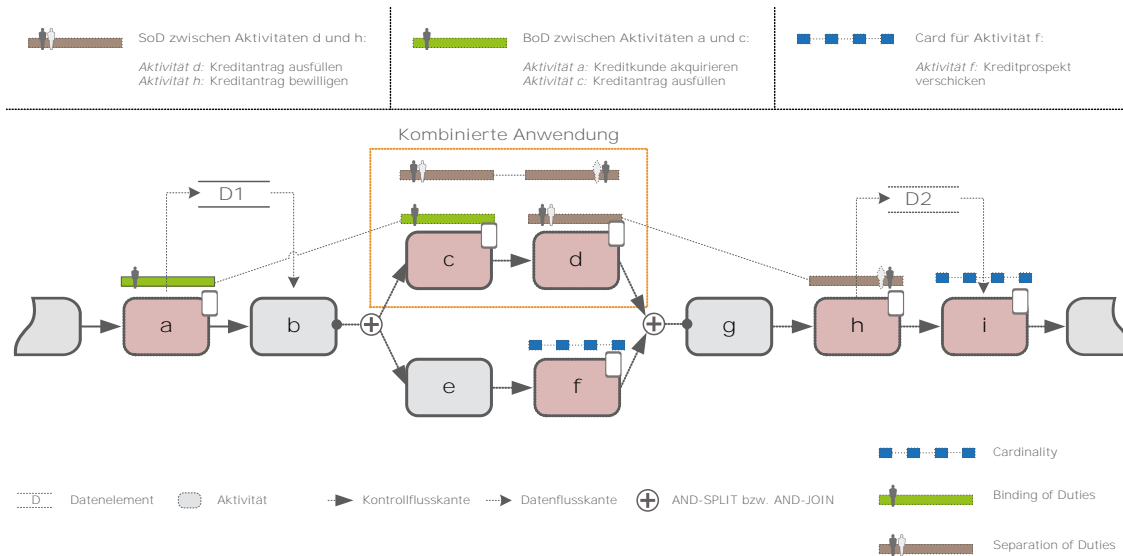


Abbildung 10.1: Kombinierte Anwendung von Prozess-Constraints

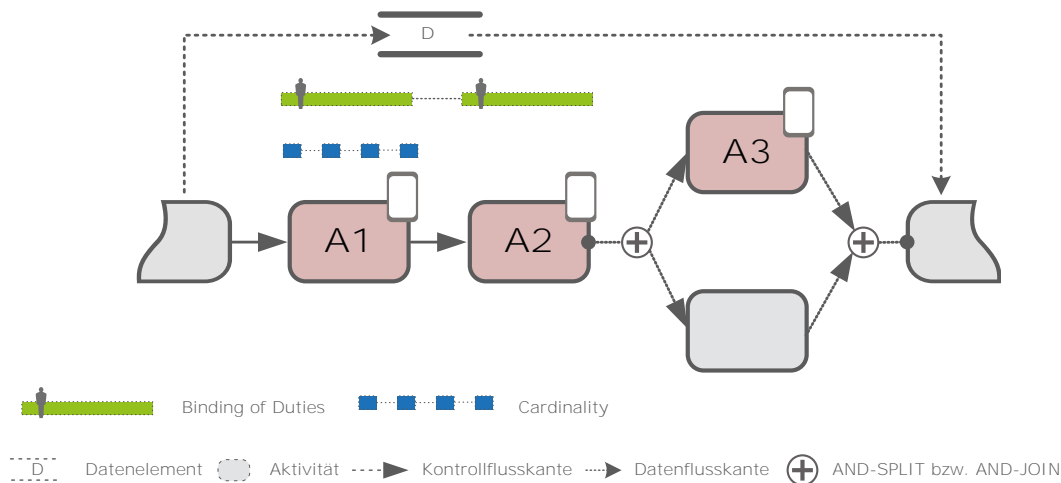


Abbildung 10.2: Kombinierte Anwendung von Card- und BoD-Constraints

mobile Endanwender der zweiten Aktivität feststeht.

Ein **BoD**-Constraint für zwei mobile Aktivitäten legt fest, dass derselbe mobile Endanwender beide mobilen Aktivitäten ausführen muss. Daher sollte die Ressourcensituation seines Smart-Mobilgeräts besonders gut sein. In Kapitel 8 wurde gezeigt, wie Algorithmus 8.2 geeignete mobile Endanwender für die Ausführung mobiler Aktivitäten berechnet. Der Algorithmus zielt primär darauf ab, die mobilen Endanwender auszuwählen, die "nahe"² der mobilen Aktivität sind. Weiters wird für alle mobilen Endanwender, die der mobilen Aktivität "nahe" sind, geprüft, welcher von ihnen die beste Ressourcensituation vorweist.

Bei der ersten mobilen Aktivität eines **BoD**-Constraints wird Algorithmus 8.2 nicht angewandt.

²D.h. die Endanwender befinden sich innerhalb des definierten Einzugsbereichs (vgl. Kapitel 7.3.2).

Kombination	Anwendungsdomäne	Betrachtetes Beispiel
BoD-BoD	Finanzwesen	Die Aktivitäten <i>Kreditantrag bearbeiten</i> , <i>Kreditantrag ausführen</i> und <i>Kreditantrag bewilligen</i> müssen vom selben Sachbearbeiter durchgeführt werden.
SoD-SoD	Gesundheitswesen	Die Aktivitäten <i>Patienten aufnehmen</i> , <i>Operationsaufklärung machen</i> und <i>Operation abrechnen</i> müssen von drei verschiedenen Ärzten durchgeführt werden.
SoD-BoD	Automobilindustrie	Die Aktivitäten <i>Bauteil ändern</i> und <i>Bauteil genehmigen</i> müssen von verschiedenen Sachbearbeitern durchgeführt werden. Weiters muss die Aktivität <i>Bauteil dokumentieren</i> vom Sachbearbeiter der Bauteilgenehmigung durchgeführt werden.
BoD-SoD	Automobilindustrie	Die Aktivitäten <i>Bauteil ändern</i> und <i>Bauteiländerung dokumentieren</i> müssen vom selben Sachbearbeiter durchgeführt werden. Weiters muss die Aktivität <i>Vorstand informieren</i> von einem anderen Sachbearbeiter durchgeführt werden.
Card-BoD	Gesundheitswesen	Die Aktivitäten <i>Blutdruck messen</i> (mehrfach) und <i>Patienten übergeben</i> müssen vom selben Sanitäter durchgeführt werden.
BoD-Card	Luftfahrt	Die Aktivitäten <i>Temperatur bestimmen</i> und <i>Temperatur kontrollieren</i> müssen vom selben Sachbearbeiter durchgeführt werden. Weiters muss auch die Aktivität <i>Temperatur kontrollieren</i> (mehrfach) vom selben Sachbearbeiter durchgeführt werden.
Card-SoD	Luftfahrt	Die Aktivitäten <i>Temperatur kontrollieren</i> (mehrfach) und <i>IT-System eintragen</i> müssen von verschiedenen Sachbearbeitern durchgeführt werden.
SoD-Card	Klinische Studien	Die Aktivitäten <i>Fragebogen ausfüllen</i> und <i>Fragebogen kontrollieren</i> (mehrfach) dürfen nicht vom selben Psychologen durchgeführt werden.

Tabelle 10.3: Praxisrelevanz einer kombinierten Anwendung von Prozess-Constraints

Der Grund dafür ist, dass der Algorithmus primär auf eine Ortsübereinstimmung abzielt, die beim **BoD**-Constraint aus drei Gründen nicht vorrangig ist. Erstens ist eine Auswertung des Ausführungsorts für die zweite mobile Aktivität bei der Zuteilung nicht mehr relevant, da dort kein weiterer mobiler Endanwender mehr ausgewählt werden muss. Zweitens müsste diese Ortsangabe bereits bei der Zuteilung der ersten mobilen Aktivität bestimmt werden, was jedoch nicht möglich ist, da der mobile Endanwender sich nur bei einer mobilen Aktivität in der Nähe aufhalten kann. Außer beide Aktivitäten werden am selben Ausführungsort durchgeführt, dann wird aber bereits durch die erste Aktivität der richtige Ausführungsort für beiden Aktivitäten bestimmt. Da der ausgewählte mobile Endanwender zwei mobile Aktivitäten ausführen muss, ist seine Ressourcensituation wichtiger einzustufen als die Tatsache, ob er sich nun nahe am Ausführungsort der mobilen Aktivität aufhält oder nicht. Da einerseits der Ausführungsort nicht bewertet werden muss und andererseits die Ressourcensituation weniger wichtig erscheint, wird anstelle von Algorithmus 8.2 der Algorithmus 8.3 angewandt. Dieser wurde entwickelt, falls Algorithmus 8.2 keine geeignete Ortsübereinstimmung findet. Da Algorithmus 8.3 primär die Ressourcensituation berücksichtigt, wird er für die erste mobile Aktivität des **BoD**-Constraints angewandt. Allerdings wird Algorithmus 8.3 zu Algorithmus 10.1 abgewandelt, um die in den

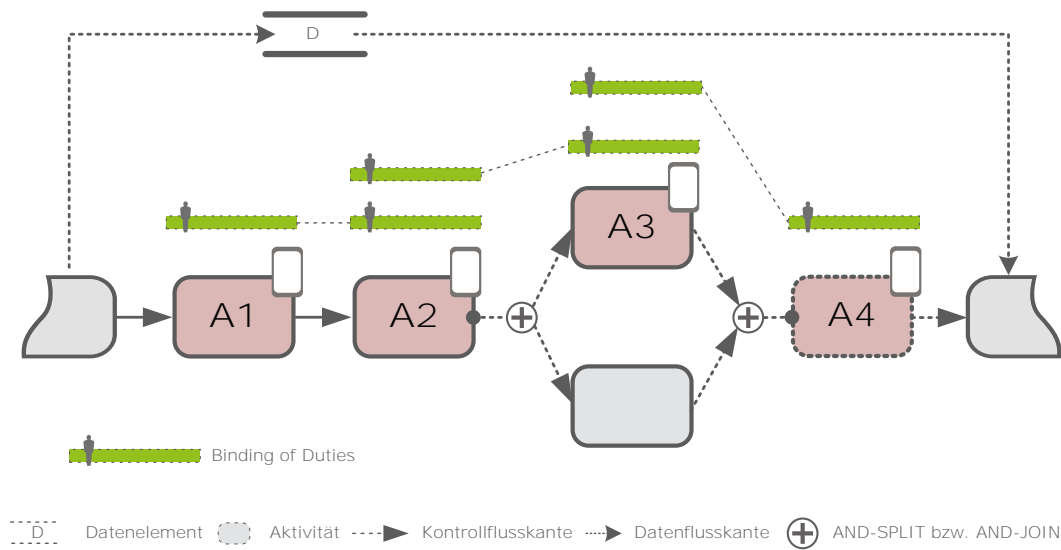


Abbildung 10.3: Kombinierte Anwendung mehrerer BoD-Constraints (BoD-Kette)

Zeilen 7-9 dargestellten Maßnahmen umzusetzen. Gilt für einen mobilen Endanwender bei der Zuteilung der ersten mobilen Aktivität weiters, dass für seine symbolischen Aufenthaltsorte eine Übereinstimmung mit dem symbolischen Ausführungsort der zweiten mobilen Aktivität gefunden werden kann, wird dieser Endanwender höher priorisiert als andere mobile Endanwender. Dadurch werden seine präferierten Ausführungsorte bei der zweiten mobilen Aktivität des **BoD**-Constraints positiv berücksichtigt.

10.3.2 Anpassung 2: Worklist-Information zu abhängiger BoD-Aktivität A2

Der mobile Endanwender, der die erste mobile Aktivität A1 des **BoD**-Constraints wählt, entscheidet sich implizit auch für die zweite mobile Aktivität A2. Daher sollte man ihm dies frühzeitig kommunizieren, sodass er weiß, dass er eine abhängige Aktivität A2 ebenfalls durchführen muss. Dazu wird ihm diese Information vor der Annahme von A1 in seiner Worklist angezeigt (vgl. Abb. 10.4 ①,②,③). Darüber hinaus wird diese Information demjenigen mobilen Endanwender angezeigt, der A1 schließlich annimmt (vgl. Abb. 10.4 ④). Auf diese Weise wird der ausführende mobile Endanwender sensibilisiert, dass die Ausführung von A1 für ihn auch implizit die Ausführung von A2 bedeutet. So kann der mobile Endanwender die mobile Aktivität bei knapper Ressourcensituation noch ablehnen.

10.3.3 Anpassung 3: Priorisierung BoD-Aktivität A1

In Abschnitt 8.3.3 wurden Regeln vorgestellt, die eine mobile Aktivität höher bzw. niedriger als andere Aktivitäten in einer Worklist priorisieren. Nun kommt für A1, die erste mobile Aktivität des **BoD**-Constraints, eine Regel hinzu, gemäß der die Priorität für A1 um 1 erhöht wird. Dadurch wird dem mobilen Endanwender vor Augen geführt, dass A1 eine höhere Priorität

A	Phase	Anpassung	Grund	A1	A2
SoD-Constraint (A1,A2)					
Es sind keine Anpassungen für den SoD-Constraint notwendig.					
BoD-Constraint (A1,A2)					
1	E	Zuteilungsschema wird geändert.	Ein mobiler Endanwender muss zwingend zwei mobile Aktivitäten durchführen und sollte daher geeignet ausgewählt werden.	✓	X
2	E	Worklist wird geändert.	Ein mobiler Endanwender muss zwingend zwei mobile Aktivitäten durchführen und sollte daher auf diesen Umstand hingewiesen werden.	✓	X
3	E	Worklist-Priorisierung wird geändert.	Ein mobiler Endanwender muss zwingend zwei mobile Aktivitäten durchführen und sollte daher auf diesen Umstand hingewiesen werden.	✓	X
4	E	Worklist wird geändert.	Ein mobiler Endanwender muss ggf. weitere mobile Aktivitäten durchführen, die zwischen A1 und A2 liegen. Darauf sollte er hingewiesen werden.	✓	X
Card-Constraint (A1)					
5	E	Zuteilungsschema wird geändert.	Ein mobiler Endanwender muss ggf. eine mobile Aktivität n-mal durchführen und sollte daher geeignet ausgewählt werden.	–	
6	E	Worklist wird geändert.	Ein mobiler Endanwender muss mehr als eine mobile Aktivität durchführen. Auf diesen Umstand sollte hingewiesen werden.	–	
7	E	Worklist-Priorisierung wird geändert.	Ein mobiler Endanwender muss mehr als eine mobile Aktivität durchführen. Auf diesen Umstand sollte hingewiesen werden.	–	
✓=trifft zu, X=trifft nicht zu, –=irrelevant, E=Erzeugung, A=Anpassung					

Tabelle 10.4: Anpassungen für Prozess-Constraints

hat, da Abhängigkeiten zu einer weiteren mobilen Aktivität A2 durch den **BoD**-Constraint bestehen.

10.3.4 Anpassung 4: Worklist-Information zu BoD-Zwischenaktivitäten

Anpassung 4 erfordert eine weitere Überlegung (vgl. Abb. 10.5). Wie Abbildung 10.5 zeigt, kann es vorkommen, dass der mobile Endanwender, der die beiden mobilen Aktivitäten A1 und A2 des **BoD**-Constraints (A1,A2) ausführt, sich auch für weitere mobile Aktivitäten qualifiziert, deren Ausführung zwischen A1 und A2 liegen kann. Für diesen Fall könnte man entscheiden, diesen mobilen Endanwender für die dazwischenliegenden mobilen Aktivitäten auszuschließen, um die Ressourcen seines Smart-Mobilgeräts zu schonen. Ferner würde dies bedeuten, dass man diesen mobilen Endanwender von mobilen Aktivitäten ausschließen müsste, die sich nicht auf diesen Prozess beziehen (d.h. Aktivitäten aus parallelen Prozessen). Jedoch ist dies aus Praxissicht nicht hilfreich, da jeder mobile Endanwender aufgrund der hohen Dynamik mobiler

Algorithmus 10.1 : Bestimmung von $KMEZL(n)$ mit **BoD** Berücksichtigung

Data : Relevante Parameterwerte des mobilen Kontexts (vgl. Tabelle 7.1)

Regeln des mobilen Kontexts (vgl. Tabelle 8.2)

 $MAME \subseteq PEA(n)$: Menge angemeldeter und berechtigter mobiler Endanwender

 $MASAO(m)$: Symbolischer Ausführungsort der zweiten mobilen Aktivität des **BoD**-Constraints

 Prozess-Constraints $EA3$ - $EA5$ der mobilen Aktivität

 $bodpw$: Dieser Wert drückt die spezifische Priorisierung aus

Result : $KMEZL(n)$: Liste geeigneter mobiler Endanwender zur Erzeugung der mobilen Aktivität n

```

1  begin
2       $KMEZL(n) \leftarrow \emptyset$ ;                                /* Liste geeigneter mobiler Endanwender initialisieren */
3      if  $|MAME| > MASWME(n)$  then
4           $KMEZLH(n) \leftarrow \emptyset$ ;                        /* Hilfsliste */
5           $EW \leftarrow \emptyset$ ;                            /* Liste für Eignungswert mobiler Endanwender initialisieren */
6           $eai \leftarrow 0$ 
7          foreach Endanwender  $ea \in MAME$  do
8              if  $EA3(n) \wedge (ME_{SAO}(ea) = MASAO(m))$  then
9                   $EW[eai] \leftarrow bodpw + SMG_{BLS}(ea) + \frac{1}{ME_{ISZ}(ea)}$ ;
10             end
11             else if  $EA3(n)$  then
12                  $EW[eai] \leftarrow SMG_{BLS}(ea) + \frac{1}{ME_{ISZ}(ea)}$ ;
13             end
14              $KMEZLH \leftarrow KMEZLH \cup \{ea\}$ ;
15              $eai++$ ;
16         end
17         /*  $KMEZLH, KMEZLH[1], \dots, KMEZLH[m]$  gemäß  $EW$  absteigend sortieren */
18          $tmp \leftarrow \emptyset$ ;                                /* TMP-Variable für einzusortierenden Wert initialisieren */
19          $tmp2 \leftarrow \emptyset$ ;                            /* TMP-Variable für einzusortierenden Wert initialisieren */
20          $j \leftarrow 0$ ;                                      /* Hilfszähler initialisieren */
21         for  $i \leftarrow 1$  to  $|KMEZLH|$  do
22              $tmp \leftarrow KMEZLH[i]$ ;
23              $tmp2 \leftarrow EW[i]$ ;
24              $j \leftarrow i$ ;
25             while  $(j > 1 \wedge KMEZLH[j-1] < EW[j])$  do
26                  $KMEZLH[j] \leftarrow KMEZLH[j-1]$ ;
27                  $EW[j] \leftarrow EW[j-1]$ ;
28                  $j \leftarrow j-1$ ;
29                  $KMEZLH[j] \leftarrow tmp$ ;
30                  $EW[j] \leftarrow tmp2$ ;
31             end
32         end
33         /* Obersten  $MASWME(n)$  Einträge aus  $KMEZLH$  in  $KMEZL(n)$  überführen */
34         for  $i \leftarrow 1$  to  $MASWME(n)$  do
35              $KMEZL(n) \leftarrow KMEZL(n) \cup KMEZLH[i]$ ;
36         end
37     end
38 else
39     foreach Endanwender  $ea \in MAME$  do
40          $KMEZL(n) \leftarrow KMEZL(n) \cup \{ea\}$ ;
41     end
42 end
    
```

Umgebungen an sich eine wertvolle Ressource ist und prinzipiell immer infrage kommen sollte. Dennoch wird eine Maßnahme vorgenommen, um diesen Kontext zu berücksichtigen. Werden

Worklist			
Meine Aktivitäten		Anstehende Aktivitäten mit Abhängigkeiten	
Aktivität	Status	Zeitlimit	Abhängigkeiten zu
Blutabnahme Frau Heller	Anstehend	4h 30min	1: Blutdruckmessung Frau Heller
.....
.....
.....
.....

Worklist		
Meine Aktivitäten		Anstehende Aktivitäten
Aktivität	Status	Zeitlimit
Blutabnahme Frau Heller	Angenommen (mit Abhängigkeiten)	4h 30min
Bestätige Erhalt Blutuntersuchung Horsch	gestartet	13:30 Uhr
Befundung Frau Zemas	angenommen
Laborwerte Herr Zweller anfordern	gestartet	0h 0min
.....

(a) Marc Schickler:
Anstehende Aktivitäten mit Abhängigkeiten

(b) Marc Schickler:
Meine Aktivitäten

Abbildung 10.4: Binding of Duties und Worklist

dem mobilen Endanwender weitere mobile Aktivitäten zur Ausführung angeboten, die zwischen den Aktivitäten eines **BoD**-Constraints liegen, wird ihm dies in der Worklist entsprechend angezeigt. D.h. der Endanwender wird darüber informiert, dass noch abhängige mobile Aktivitäten des **BoD**-Constraints auszuführen sind. So kann der mobile Endanwender die mobile Aktivität bei knapper Ressourcensituation noch ablehnen.

10.3.5 Anpassung 5: Zuteilungsschema Card-Aktivität

Anpassung 5 bezieht sich auf den **Card**-Constraint. Bei diesem sollte einerseits die Nähe des mobilen Endanwenders zur mobilen Aktivität berücksichtigt werden, andererseits sollte derjenige Endanwender ausgewählt werden, der die beste Ressourcensituation vorweist. Prinzipiell ist eine solche Auswahl durch Algorithmus 8.2 bereits sichergestellt; dieser kommt standardmäßig für die Zuteilung mobiler Aktivitäten zu mobilen Endanwendern zur Anwendung.

Nun wird noch eine weitere Maßnahme durchgeführt. Der gesetzte Schwellwert einer mobilen Aktivität wird nicht mehr berücksichtigt und die geometrische Koordinatenübereinstimmung wird nicht mehr höher priorisiert als die symbolische Koordinatenübereinstimmung. Beibehalten wird die Bewertung der Ressourcensituation der mobilen Endanwender. Die Menge an geeigneten mobilen Endanwendern wird dadurch im Regelfall größer als bei Anwendung von Algorithmus 8.2. Im Gegenzug wird die Nähe der mobilen Endanwender zur mobilen Aktivität weniger berücksichtigt, da die geometrische Koordinatenübereinstimmung nicht mehr höher als die symbolische Koordinatenübereinstimmung bewertet wird. Da ggf. viele Ausführungen der

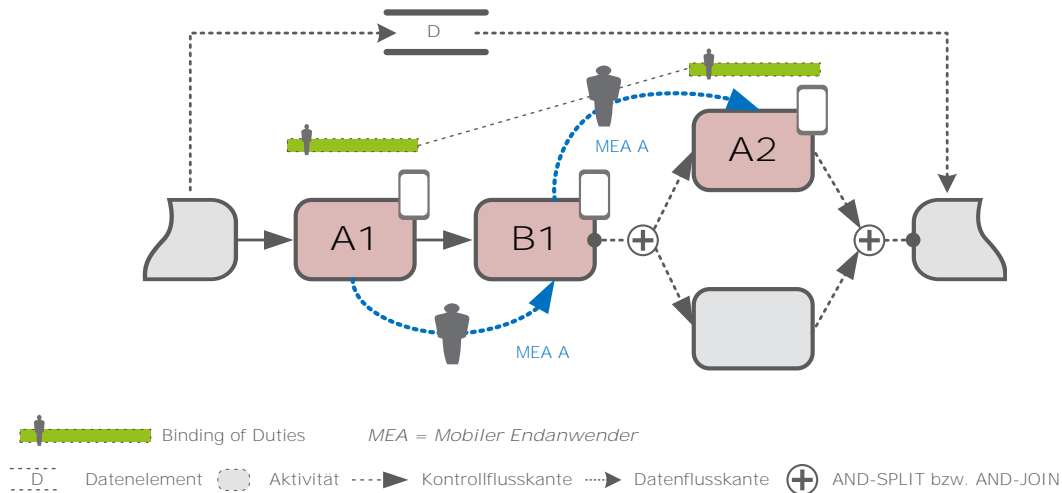


Abbildung 10.5: Binding of Duties mit dazwischenliegender mobiler Aktivität

mobilen Aktivität aufgrund des **Card-Constraints** erfolgen, ist die Koordinatenübereinstimmung weniger wichtig als die Ressourcensituation.

Auch hier kommt wieder Algorithmus 8.3 infrage, da er die Ressourcensituation ebenfalls vorrangig berücksichtigt. Jedoch kommt er nicht zum Einsatz, da die Koordinatenübereinstimmung durch diesen Algorithmus nicht berücksichtigt wird.

10.3.6 Anpassungen 6-7: Worklist-Priorisierung von Card-Aktivitäten

Die Anpassungen 6 und 7 beziehen sich ebenfalls auf den **Card-Constraint**. Sie erfolgen analog zu den Anpassungen 2 und 3 des **BoD-Constraints**. Bei einem **Card-Constraint** muss der mobile Endanwender mehrere mobile Aktivitäten zwingend ausführen. Dies sollte einem mobilen Endanwender entsprechend angezeigt werden, bevor er die mobile Aktivität annimmt. Ferner sollte eine solche mobile Aktivität in der Worklist höher priorisiert werden. Daher wird bei Anpassung 6 analog zur Anpassung 2 eine Anzeige in der Worklist realisiert. Entsprechend wird bei Anpassung 7 analog zur Anpassung 3 eine Regel eingeführt, welche die Priorität einer mobilen Aktivität mit **Card-Constraint** um 1 erhöht.

10.3.7 Anpassungen bei kombinierter Anwendung der Prozess-Constraints

Im Folgenden wird diskutiert, ob bei kombinierter Anwendung der Prozess-Constraints weitere Anpassungen erforderlich sind. Wie bei den Anpassungen für den einfachen Fall gezeigt wurde, werden bei der Erzeugung und Ausführung mobiler Aktivitäten nur für die **BoD-** und **Card** Constraint-Kombinationen Änderungen vorgenommen. Die Änderungen bestehen einerseits aus einer Anwendung der bereits vorgestellten Anpassungen 1 bis 7. Ferner werden weitere Anpassungen vorgenommen, (s. Tabelle 10.5).

A	Phase	Anpassung	A1	A2	A3
BoD-BoD-Constraint-Kombination (A1,A2) u. (A2,A3)					
1	E	Algorithmus 10.1 wird derart angepasst, dass nicht nur der symbolische Ausführungsort von A2, sondern auch der symbolische Ausführungsort von A3 berücksichtigt wird.	✓	X	X
2	E	Da ein mobiler Endanwender durch diese Constraint-Kombination drei mobile Aktivitäten ausführen muss, wird ihm bei A1 nicht nur die Abhängigkeit zu A2, sondern auch diejenige zu A3 angezeigt.	✓	X	X
3	E	Da ein mobiler Endanwender durch diese Constraint-Kombination drei mobile Aktivitäten ausführen muss, wird eine neue Prioritätsregel angewandt, die bei A1 die Priorität um 2 (da noch zwei abhängige Aktivitäten folgen) und bei A2 die Priorität um 1 (da noch eine abhängige Aktivität folgt) in seiner Worklist erhöht.	✓	X	X
Card-BoD-Constraint-Kombination (A1,A2)					
6	E	Es wird einerseits die mehrfache Ausführung von Aktivität A1 angezeigt und andererseits die Abhängigkeit zur Aktivität A2.	✓	–	
7	E	Bei A1 wird durch eine weitere Regel die Priorität um 2 erhöht.	✓	–	
BoD-Card-Constraint-Kombination (A1,A2)					
Die BoD-Card-Constraint-Kombination wird wie eine BoD-BoD-Constraint-Kombination behandelt.					
✓=trifft zu, X=trifft nicht zu, –=irrelevant, E=Erzeugung, A=Anpassung					

Tabelle 10.5: Anpassungen für kombinierte Anwendung der Prozess-Constraints

10.4 Anpassungen bei der Ausnahmebehandlung mobiler Aktivitäten

In Abschnitt 10.3 wurden die Anpassungen vorgestellt, die im Zusammenhang mit Prozess-Constraints bei der Erzeugung und Ausführung mobiler Aktivitäten vorzunehmen sind. Nun werden die Anpassungen diskutiert, die zwecks Sicherstellung der Prozess-Constraints im Kontext der in Kapitel 9 vorgestellten Ausnahmebehandlung erforderlich werden. Tabelle 10.6 stellt diese im Überblick vor.

10.4.1 Anpassung 8: Zuteilungsschema SoD-Aktivität A1

Die Ausnahmebehandlung bei Vorliegen eines *SoD*-Constraints erfordert bezogen auf eine Delegation weitergehende Überlegungen (s. Abb. 10.6). In Abbildung 10.6 ist im roten Kasten (*vor Optimierung*) ein möglicher Delegationsfall für die beiden Aktivitäten mit *SoD*-Constraint – A1 und A2 – gezeigt. Algorithmus 9.1 priorisiert bei A1 den mobilen Endanwender A am höchsten, den mobilen Endanwender B am zweithöchsten und schließlich den mobilen Endanwender C an dritter Stelle der Delegationsliste. Wird nun angenommen, dass bei A1 zwei Delegationen zu den mobilen Endanwendern A und B auftreten und erst der mobile Endanwender B die mobile Aktivität ordnungsgemäß ausführt, würden bei dem in Abbildung 10.6 dargestellten Szenario die mobilen Endanwender A und B auch für A2 nicht mehr infrage kommen. Sie kommen deshalb nicht mehr infrage, weil sie bereits an der Ausführung von A1 beteiligt waren und somit bei

A	Anpassung	Grund	A1	A2
SoD-Constraint (A1,A2)				
8	Zuteilungsschema für Delegation wird geändert	Die Delegation sollte mobile Endanwender, die für A1 und A2 infrage kommen, bei A1 möglichst ausschließen.	✓	X
9	Ausnahmebehandlung wird geändert	Das Backup muss den <i>SoD</i> -Constraint ebenfalls berücksichtigen.	✓	✓
BoD-Constraint (A1,A2)				
10	Zuteilungsschema für Delegation wird geändert	Die Delegation sollte den Ausführungsort von A2 berücksichtigen.	✓	X
11	Ausnahmebehandlung für A1 wird geändert	Da die Ausnahmebehandlung von A1 auch die Ausnahmebehandlung von A2 bestimmt, muss die Auswahl der Ausnahmebehandlung bei A1 bereits die geeignete Ausnahmebehandlung von A2 mit berücksichtigen.	✓	X
12	Ausnahmebehandlung für A2 wird geändert	Die Ausnahmebehandlung von A2 wird auf die Ausnahmebehandlung von A1 abgestimmt.	X	✓
Card-Constraint (A1)				
13	Ausnahmebehandlung wird geändert	Die Ausnahmebehandlung wird gemäß der Anpassung 5 (vgl. Abschnitt 10.3.5) bzw. gemäß den Anpassungen 6 und 7 (vgl. Abschnitt 10.3.6) verändert.	X	✓
✓=trifft zu, X=trifft nicht zu, –=irrelevant, A=Anpassung				

Tabelle 10.6: Anpassungen für die Prozess-Constraints bei der Ausnahmebehandlung

einer Ausführung von A2 der *SoD*-Constraint verletzt wird, wenn diese mobilen Endanwender ausgewählt werden.

Für A2 käme im gezeigten Szenario lediglich der mobile Endanwender D für eine Delegation infrage. Sortiert man die Liste bei A1 jedoch um und nimmt wieder dieselbe Anzahl an Delegationen vor, kämen (siehe den grünen Kasten; *nach Optimierung*) weitere zwei mobile Endanwender für eine Delegation von A2 infrage. Daher wird die Delegationsliste bei Vorliegen eines *SoD*-Constraints speziell berechnet. Diese Berechnung wiederum zeigt Algorithmus 10.2. Hier ist Zeile 7 entscheidend: sie regelt, dass mobile Endanwender, die bei A1 auch für A2 infrage kommen, schlechter priorisiert werden. Dazu wird der Priorisierungswert ME_{DPV} nochmals mit *sodpw* addiert.³

An dieser Stelle soll noch eine weitere mögliche Maßnahme diskutiert werden. Prinzipiell könnte anstatt einer Anpassung des Zuteilungsschemas auch eine algorithmische Analyse untersuchen, ob nach einer Delegation bei A1 der *SoD*-Constraint bei A2 weiter erfüllbar bleibt. Delegation im Kontext des *SoD*-Constraints bedeutet, dass ein mobiler Endanwender aus der möglichen Liste der mobilen Endanwender herausfällt. Verwandte Ansätze beschäftigen sich mit solchen Algorithmen sehr intensiv. Jedoch sind diese Berechnungen sehr aufwändig, vor allem weil sie praktisch bei jeder Delegation zur Laufzeit eines Prozesses neu angewandt werden müssen. Ferner können Constraints wiederum weitere Constraints beeinflussen, sodass nicht nur der konkret

³Zur Erinnerung: Elemente mit kleinem ME_{DPV} haben einen besseren Wert. *sodpw* wird wiederum aus umfangreichen Praxistests festgelegt.

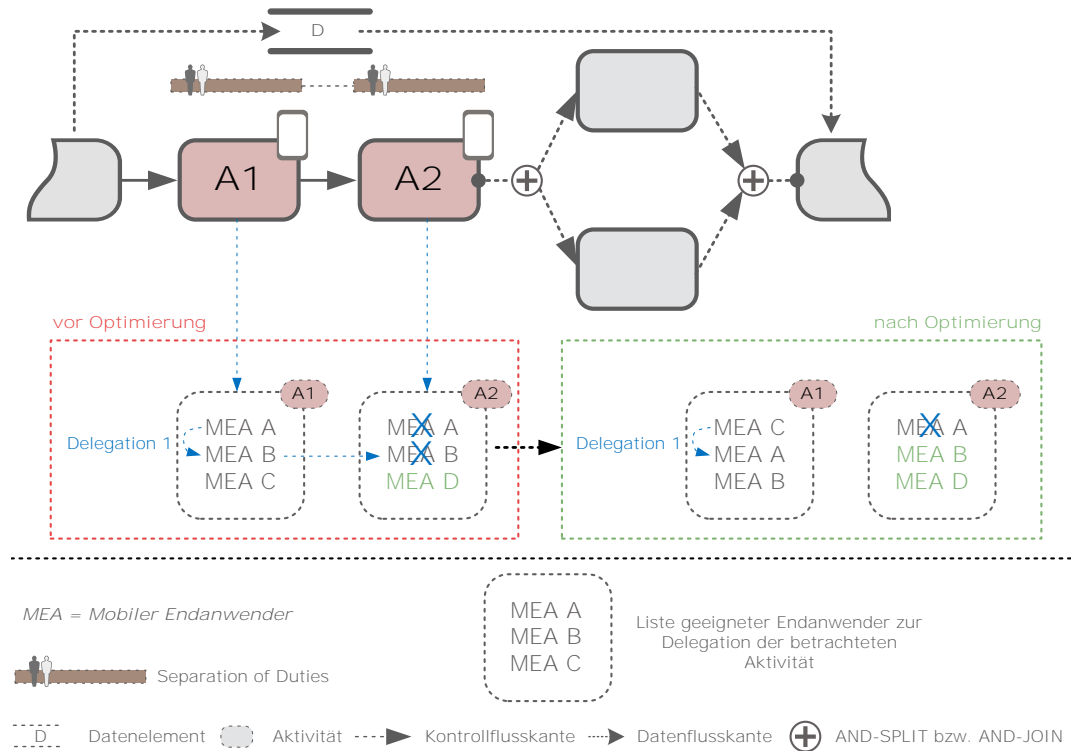


Abbildung 10.6: Ausnahmebehandlung bei Vorliegen eines *SoD*-Constraints

anstehende Constraint betrachtet werden muss, sondern auch die abhängigen mit einzubeziehen sind.

Stellt man sich nun ein *PrMS* mit vielen parallel laufenden Instanzen vor⁴, kann dieses durch Berechnungen extrem belasten werden. Dies soll durch ein Beispiel illustriert werden. In [Cra04] wird ein Algorithmus vorgestellt, der eine solche Betrachtung ermöglicht. Dieser Algorithmus betrachtet einen Prozess, eine Menge von Constraints auf einem Prozess und eine Menge von Endanwendern, die jeder Prozessaktivität zugeordnet sind.

Nun kann mithilfe des Algorithmus aus [Cra04] berechnet werden, ob die Modellierung zu mindestens einer korrekten Ausführung führt. Ändert sich dadurch die Anzahl der mobilen Endanwender zur Laufzeit des Prozesses, muss der Algorithmus entsprechend nochmals ausgeführt werden. Dieses Szenario entspricht der Delegation beim *SoD*-Constraint.

Nun soll das Laufzeitverhalten dieses Algorithmus kurz betrachtet werden. Die Laufzeitbetrachtung ist durch Formel 10.1 beschrieben.

$$\mathcal{O}(|A|^\omega * |A|^2 * |EA|^4) \quad (10.1)$$

A bezeichnet die Aktivitäten des Prozesses, ω den Verzweigungsgrad⁵ des Prozesses und EA die Menge der berechtigten Endanwender aller Aktivitäten. Betrachtet man zum Beispiel einen Prozess mit 30 Aktivitäten, einem Verzweigungsgrad von 4 und einer Menge zu betrachtender Endanwender von 15, dies entspricht einer sehr realistischen Annahme aus der Praxis,

⁴Manches Mal bis zu tausend Instanzen in paralleler Ausführung.

⁵Der Verzweigungsgrad gibt die maximale Rekursionstiefe von Verzweigungen an.

lautet die Berechnung zu Formel 10.1: $30^4 * 30^2 * 15^4$. Das konkrete Ergebnis dazu lautet: 3388231620000000000. Bei einem aktuellen Intel Prozessor der Klasse *i7* [FPO14] und unterstellten 102,5 GFLOPS, würde ein solcher Prozessor für die Berechnung **382,59** Tage benötigen. Dennoch können solche Berechnungen dazu verwendet werden, um eine Modellierung von Prozess-Constraints vor der Ausführung eines Prozesses zu analysieren. Dieser Ansatz wurde bisher aus Praxissicht nicht weiter verfolgt, da sich die Anpassung durch Algorithmus 10.2 bewährt hat.

10.4.2 Anpassung 9: Zuteilungsschema Backup SoD-Aktivität A2

Ein weiterer Aspekt ergibt sich durch das Backup mobiler Aktivitäten. Wird bei den mobilen Aktivitäten A1 und A2 das Backup notwendig, darf der Endanwender von A1 nicht das Backup von A2 durchführen. Grundsätzlich werden über diesen Aspekt hinaus auch das Abbrechen und das Backup so angewandt, als ob kein *SoD*-Constraint vorliegen würde. Dies betrifft auch den Systemwechsel.

10.4.3 Anpassung 10: Zuteilungsschema Delegation BoD-Aktivität A1

Der Algorithmus zur Bestimmung der Delegationsliste der mobilen Aktivität A1 muss in Verbindung mit dem **BoD**-Constraint geändert werden. Bereits bei der Erzeugung mobiler Aktivitäten mit **BoD**-Constraint wurde diskutiert (vgl. Abschnitt 10.3.1), dass das Zuteilungsschema für A1 geändert werden muss. Nur so wird man der Situation gerecht, dass ein mobiler Endanwender zwei mobile Aktivitäten ausführen muss. Für die Delegation muss dieser Umstand abermals berücksichtigt werden. Dabei muss der Ausführungsort von Aktivität A2 für die Delegationsliste berücksichtigt werden. Für diesen Zweck wird Algorithmus 10.3 eingeführt, der eine Kombination der Algorithmen 9.1 und 10.1 realisiert. Einerseits berücksichtigt Algorithmus 10.3 (vgl. Zeile 7) den Ausführungsort von A2, andererseits die relevanten Kontextparameter, die bereits bei deren Delegation (ohne **BoD**-Constraint) vorgestellt wurden (vgl. Zeilen 8 und 11 in Algorithmus 10.3).

10.4.4 Anpassung 11: Ausnahmebehandlung BoD-Aktivität A1

Die Ausnahmebehandlung für Aktivität A1 eines **BoD**-Constraints (A1,A2) muss in jedem Fall geändert werden. Dazu wiederholen wir kurz das Prozedere der Ausnahmebehandlung für A1: Zuerst wird eine Delegation für A1 vorgenommen. Dazu werden geeignete mobile Endanwender als Ziele der Delegation durch Algorithmus 10.3 bestimmt. Sollte die Delegation nicht möglich sein, wird A1 abgebrochen, sofern es sich um eine optionale mobile Aktivität handelt. Andernfalls kommt die Backupvariante BV1 zur Anwendung, d.h. es liegt eine kritische mobile Aktivität vor.

Zu beantworten ist noch die Frage, was diese beiden Fälle für die Ausführung der mobilen Aktivität A2 bedeuten. Sollte A1 abgebrochen werden, verbleiben zwei Möglichkeiten für die Ausführung von A2 unter Beibehaltung des **BoD**-Constraints: entweder wird A2 ebenfalls abgebrochen oder aber wird A2 demjenigen mobilen Endanwender zugeteilt, der die Delegation bei A1 zuletzt nicht durchführen konnte. Das letztere Vorgehen verletzt den **BoD**-Constraint nicht. Jedoch muss einkalkuliert werden, dass der mobile Endanwender, der die letzte Delega-

tion bei A1 verursachte, immer noch nicht verfügbar ist. Sollte das Backup für A1 angewandt werden, verbleiben für A2 nur zwei Möglichkeiten. Ist bei A1 das Backup von einem Endanwender ausgeführt worden, der grundsätzlich keine mobilen Aktivitäten ausführt, muss A2 ebenfalls durch ein automatisches Backup am Desktop-System ausgeführt werden, da sonst der **BoD**-Constraint verletzt wird. Hat das Backup für A1 einen Endanwender ausgewählt, der prinzipiell auch mobile Aktivitäten ausführt, kann bei Aktivität A2 versucht werden, diese dem ausführenden Endanwender von A1 mobil zuzuteilen. Scheitert dieser Versuch oder verursacht der mobile Endanwender bei A2 eine Ausnahme, muss A2 ebenfalls an einem Desktop-System ausgeführt werden. Als weitere Maßnahme könnte im Fall eines notwendigen Backups noch die Validations-Aktivität hinzugefügt werden, die von demjenigen mobilen Endanwender bestätigt werden muss, der zuletzt die Ausführung innehatte.

Eine letzte Möglichkeit besteht darin, dem mobilen Endanwender, der zuletzt die Delegation annimmt, im Ausnahmefall einen Systemwechsel vorzuschlagen. Akzeptiert er diesen Systemwechsel, kann bei A1 im Ausnahmefall ein Systemwechsel durchgeführt werden und A2 kann von diesem mobilen Endanwender wieder normal ausgeführt werden, sofern dieser seinen Ausnahmezustand behoben hat. Sollte dieser mobile Endanwender für A2 nicht mobil verfügbar sein, kann abermals ein Systemwechsel automatisiert durchgeführt werden, sodass der **BoD**-Constraint sichergestellt bleibt. Daher wird bei demjenigen mobilen Endanwender, der das finale Delegationsziel darstellt, automatisch erfragt, ob er einem Systemwechsel sowohl für A1 als auch für A2 zustimmt. Bejaht er dies, wird die Ausnahmebehandlung entsprechend durchgeführt. Lehnt er den Systemwechsel ab, kommt das im Kontext von Anpassung 9 beschriebene Prozedere zum Einsatz.

10.4.5 Anpassung 12: Ausnahmebehandlung BoD-Aktivität A2

Die Anpassung 12 erfolgt analog zu den Ausführungen von Anpassung 11. Je nachdem was bei Anpassung 10 für Aktivität A1 erfolgt, wird analog bei A2 vorgegangen.

10.4.6 Anpassung 13: Ausnahmebehandlung Card-Aktivität

Die Listenbestimmung geeigneter mobiler Endanwender, die bei einer regulären Delegation erfolgt, wird bei Vorliegen eines **Card**-Constraints übernommen. Ebenso wird das Abbrechen mobiler Aktivitäten mit **Card**-Constraint bzw. die Anwendung des Backups übernommen. Die entscheidende Veränderung bei der Ausnahmebehandlung einer mobilen Aktivität mit **Card**-Constraint ist, dass Delegationen noch nach der ersten Ausführung möglich sind, sofern weitere Ausführungen anstehen. Eigentlich besagt der **Card**-Constraint, dass alle Ausführungen vom selben mobilen Endanwender ausgeführt werden sollen. Jedoch hat sich in der Praxis gezeigt, dass hier eine Übernahme durch einen weiteren mobilen Endanwender sinnvoll sein kann. Aus diesem Grund kann auch noch nach der ersten Ausführung eine Delegation erfolgen. Alternativ kann nach der ersten Ausführung beim mobilen Endanwender angefragt werden, ob er einem Systemwechsel zustimmt. Dieses Prozedere wird standardmäßig nach der ersten Ausführung angewandt. Stimmt der mobile Endanwender diesem Vorgehen zu, wird die weitere Ausnahmebehandlung obsolet.

Sollte das Abbrechen oder das Backup für A2 notwendig werden, wird folgendermaßen vorgegangen: Ist ein Abbruch erforderlich, werden alle weiteren Ausführungen abgebrochen. Sollte dagegen das Backup zur Anwendung kommen, werden dadurch alle weiteren Ausführungen

bewerkstelligt.

Als weitere Maßnahme könnte im Fall eines notwendigen Backups noch die Validations-Aktivität hinzugefügt werden. Diese muss von demjenigen mobilen Endanwender bestätigt werden, der die mobile Ausführung zuletzt innehatte.

10.4.7 Anpassung 14: Ausnahmebehandlung Offline-Aktivitäten

Die Ausnahmebehandlung für Offline-Aktivitäten erfordert ebenfalls eine Anpassung. Grundsätzlich haben Prozess-Constraints zwar keine Auswirkungen auf eine Offline-Aktivität und können daher mit mobilen Aktivitäten, für die Prozess-Constraints definiert sind, kombiniert werden. Allerdings erfordert ein konkreter Fall eine Anpassung: Wurde eine Offline-Aktivität für eine mobile Aktivität definiert, für die auch ein **Card**-Constraint vorliegt, erfolgt eine Anpassung der Ausnahmebehandlung.

Da ein **Card**-Constraint ausdrückt, dass mehrere Ausführungen nötig sind, wird die Zeit bis zum Start der Ausnahmebehandlung wie folgt berechnet. Handelt es sich um eine Offline-Aktivität ohne **Card**-Constraint, wird die Ausnahmebehandlung nach Verstreichen von *atimeout* gestartet (vgl. Abschnitt 9.7.1). Bei einer Offline-Aktivität mit **Card**-Constraint dagegen wird die Ausnahmebehandlung nach $atimeout * IterationenCardConstraint$ gestartet. D.h. für jede Iteration wird die Zeit *atimeout* für die Offline-Ausführung toleriert. Damit kann die Kombination aus **Card**-Constraint und Offline-Ausführung länger andauern.

10.4.8 Anpassungen bei kombinierter Anwendung der Prozess-Constraints

Abschließend wird diskutiert, ob weitere Anpassungen bei der Ausnahmebehandlung notwendig werden, wenn die Prozess-Constraints kombiniert angewandt werden. Die bereits vorgestellten Anpassungen 8 bis 13 gelten auch bei kombinierter Anwendung der Prozess-Constraints. Zusätzlich sind für die in Tabelle 10.7 gezeigten kombinierten Anwendungen von Prozess-Constraints weitere Änderungen der bereits vorgestellten Anpassungen erforderlich (vgl. Tabelle 10.7).

10.5 Weitere Aspekte

Dieser Abschnitt diskutiert weitere relevante Aspekte bei der Behandlung von Prozess-Constraints in Verbindung mit mobilen Aktivitäten.

10.5.1 Berücksichtigung weiterer Prozess-Constraints

Im Kontext von Prozess-Constraints und mobilen Aktivitäten konnten noch weitere, praxis-relevante Szenarien identifiziert werden, die in dieser Arbeit jedoch im Sinne einer besseren Fokussierung nicht berücksichtigt werden. Diese werden im Folgenden erwähnt, auch um die Relevanz von Prozess-Constraints im Kontext mobiler Aktivitäten zu unterstreichen. Abbildung 10.8 zeigt ein Szenario mit einem **SoD**-Constraint bezogen auf eine mobile und eine nicht

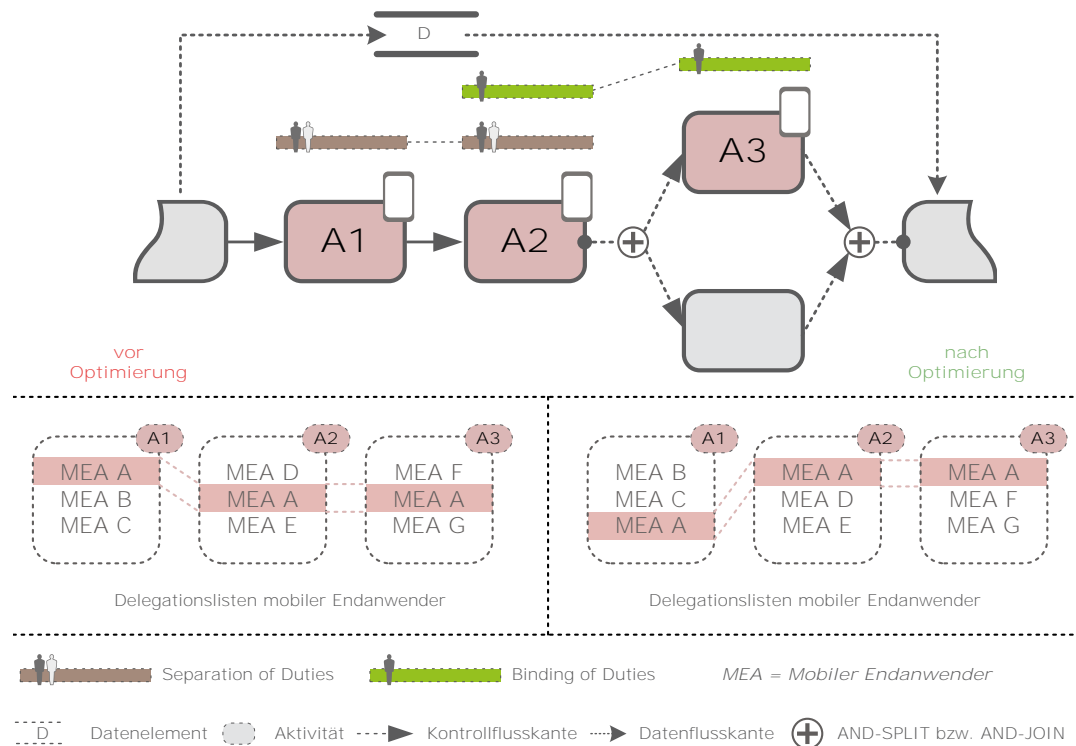


Abbildung 10.7: Delegationslistenanpassung bei kombiniertem *SoD*-Constraint

mobile Aktivität. Dieser Fall ist ebenfalls praxisrelevant, erfordert aber zusätzliche Betrachtungen. So muss explizit zwischen mobilen und nicht mobilen Endanwendern unterschieden werden, was in dieser Arbeit aber nicht erfolgt ist.

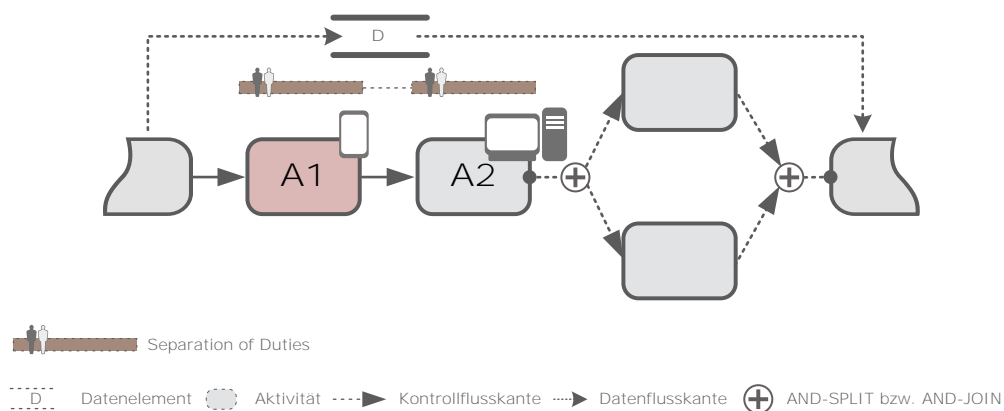


Abbildung 10.8: Prozess-Constraints und nicht mobile Aktivitäten

Abbildung 10.9 zeigt drei weitere Szenarien, die sich von den bisherigen Prozess-Constraints unterscheiden. Im Szenario I wird zwischen zwei mobilen Aktivitäten ein Prozess-Constraint definiert. Dieser drückt aus, dass sich die ausführenden Endanwender im selben Aufenthalts-

bereich befinden müssen. Im Szenario II wird der Prozess-Constraint weiter verschärft: hier müssen sich die beiden ausführenden Endanwender zwingend am identischen Ausführungsort aufhalten. Im Szenario III wird schließlich ein **SoD**-Constraint definiert. Weiters muss gelten, dass sich die ausführenden mobilen Endanwender von A1 und A2 an einem identischen Ausführungsort aufhalten müssen. Der zuletzt genannte Prozess-Constraint verknüpft die in dieser Arbeit diskutierten Prozess-Constraints mit den in den Szenarien I und II diskutierten weiteren Prozess-Constraints. Diese Szenarien unterstreichen, dass für mobile Aktivitäten durchaus noch weitere praxisrelevante Prozess-Constraints identifiziert werden können.

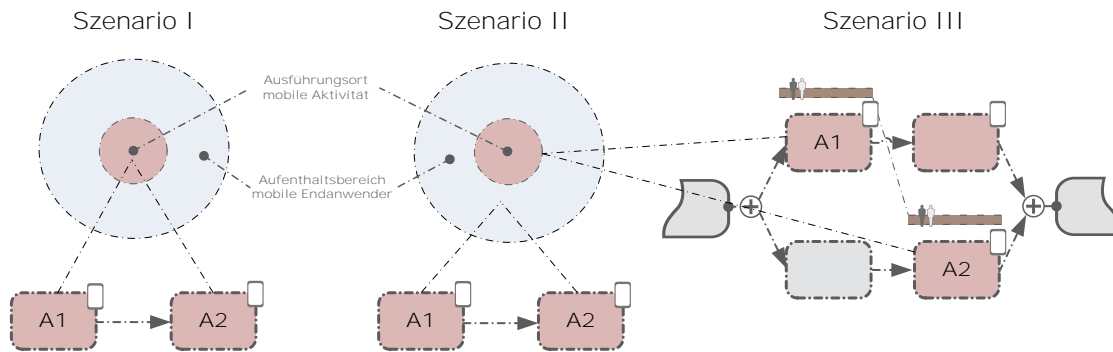


Abbildung 10.9: Weitere Prozess-Constraints im Kontext mobiler Aktivitäten

10.5.2 Inkonsistente Constraint-Modellierung

Der Aspekt einer inkonsistenten Modellierung von Prozess-Constraints wird in dieser Arbeit nicht betrachtet. So könnten zwei Constraints gleichzeitig auf zwei mobilen Aktivitäten definiert werden, etwa ein **SoD**-Constraint auf den Aktivitäten A1 und A2 und gleichzeitig ein **BoD**-Constraint auf diesen Aktivitäten. Eine solche Modellierung ist inkonsistent. Inkonsistenzen können wiederum die Sicherheit des *PrMS* gefährden [WMM09]. Um Inkonsistenzen automatisch zu erkennen, können zum Beispiel Techniken des *Model Checkings* eingesetzt werden [MC114, CGP99, SLS06, FGP⁺03, BCC⁺99, CBRZ01, WSM08, WS07, Ly13, LRMGD09, KRL⁺13].

10.6 Diskussion

Dieser Abschnitt diskutiert den Beitrag von Prozess-Constraints zur robusten Ausführung mobiler Aktivitäten und führt ferner eine Abgrenzung gegenüber verwandten Ansätzen durch.

10.6.1 Beitrag zur robusten Ausführung mobiler Aktivitäten

Die diskutierten Prozess-Constraints erfüllen mehrere Anforderungen dieser Arbeit (vgl. Kapitel 2). Tabelle 10.8 fasst diese zusammen und zeigt, welche Maßnahmen zur Erfüllung der jeweiligen Anforderung beitragen.

Abbildung 10.10 zeigt, wie gut die Konzepte zur Erzeugung und Ausführung mobiler Aktivitäten mit Prozess-Constraints kombiniert angewandt werden können. Dasselbe gilt in Bezug auf die vorgestellte Ausnahmebehandlungskonzepte für mobile Aktivitäten. Diese kombinierte Anwendung fördert die robuste Ausführung und existiert in verwandten Ansätzen so bisher nicht.

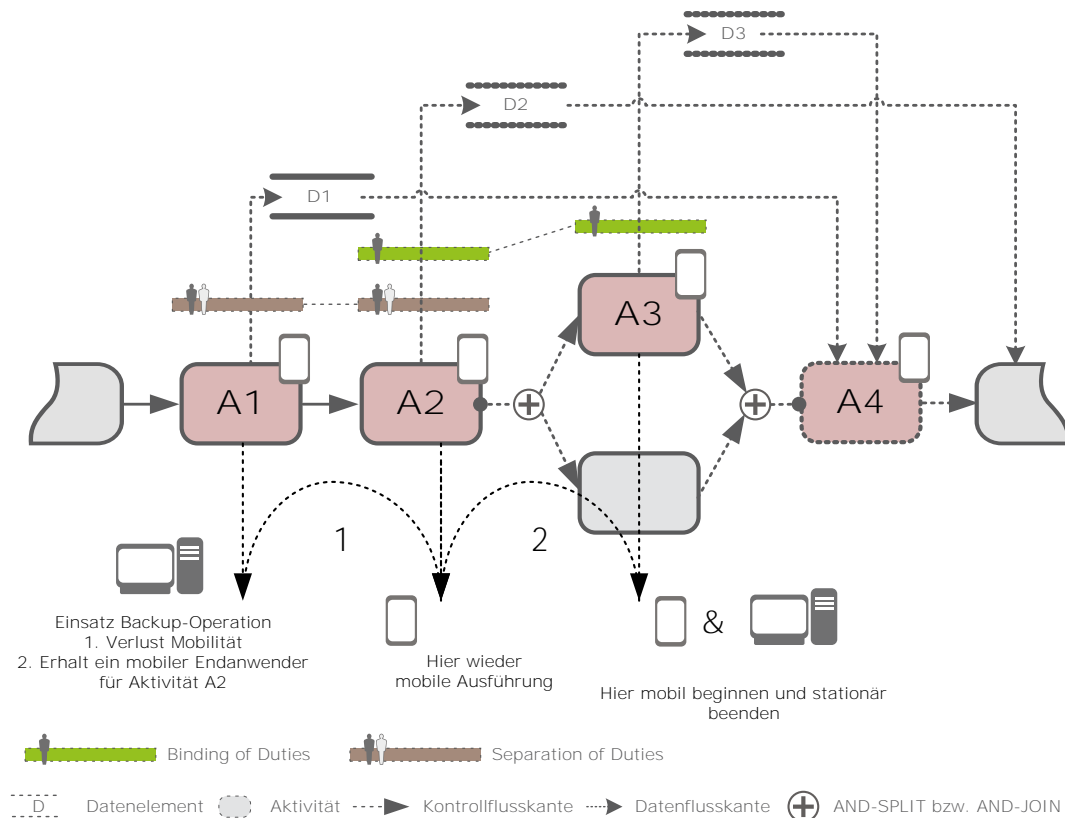


Abbildung 10.10: Kombinierte Anwendung der Konzepte

10.6.2 Verwandte Ansätze

Prozess-Constraints werden im Kontext mobiler Aktivitäten bisher kaum berücksichtigt. In [HO03] werden zwar mobile Aktivitäten und Prozess-Constraints diskutiert, allerdings handelt es sich hierbei um andersartige Prozess-Constraints. Diese Arbeit fokussiert insbesondere auf den Aspekt der Sicherheit.

Weiters wurden die Arbeiten [BFA97, BFA99, KM03, KM04, HA99, CFL04, CFL03, YAE09] analysiert, die zur grundsätzlichen Betrachtung von Prozess-Constraints im Kontext von *PrMS* relevant sind. [BFA99] gibt den umfassendsten Überblick in das Themengebiet.

Die Arbeiten aus [CFL04, CFL03] beschäftigen sich mit Constraints im Kontext von Agenten [WPA14]. Mobile Aktivitäten ohne Prozesskontext werden häufig über Agentenkonzepte realisiert, daher sind diese Ansätze für die vorliegende Arbeit relevant. Allerdings diskutieren [CFL04, CFL03] dynamische Änderungen eines Agenten im Kontext von Constraints. So wird Java-Code eines Agenten dynamisch verändert und die Constraints müssen dynamisch über-

prüft bzw. angepasst werden, wenn sich die Codebasis ändert. Insgesamt werden Prozess-Constraints im Kontext mobiler Aktivitäten bisher kaum betrachtet. Ebenso wenig werden Prozess-Constraints in Kombination mit anderen Konzepten, wie der Ausnahmebehandlung mobiler Aktivitäten, betrachtet.

10.7 Zusammenfassung

In diesem Kapitel wurden drei grundlegende Prozess-Constraints und deren kombinierte Anwendung im Kontext mobiler Aktivitäten vorgestellt. Es wurde gezeigt, wie Prozess-Constraints bezogen auf mobile Aktivitäten geeignet realisiert werden können. Speziell wurde diskutiert, wie deren Realisierung im Kontext der Erzeugung, Ausführung und Ausnahmebehandlung mobiler Aktivitäten erfolgt. Weiters wurde gezeigt, welche Herausforderungen dabei resultieren und wie diese bewerkstelligt werden können. Schließlich wurde gezeigt, dass Prozess-Constraints im Kontext mobiler Aktivitäten praxisrelevant sind und daher grundlegend zu berücksichtigen sind.

Algorithmus 10.2 : Bestimmung von $KMEZL(n)$ bei Delegationen mit SoD Berücksichtigung**Data** : Relevante Parameterwerte des mobilen Kontexts (vgl. Tabelle 7.1)

Regeln des mobilen Kontexts (vgl. Tabelle 8.2)

 $MAME \subseteq PEA(n)$: Menge angemeldeter und berechtigter mobiler Endanwender der mobilen Aktivität n $MAME2 \subseteq PEA(m)$: Menge angemeldeter und berechtigter mobiler Endanwender der mobilen Aktivität mProzess-Constraints $EA3$ - $EA5$ der mobilen Aktivität n $sodpw$: Dieser Wert drückt die spezifische Priorisierung aus**Result** : $KMEZL(n)$: Liste geeigneter mobiler Endanwender zur Delegation der mobilen Aktivität n

```

1 begin
2    $KMEZL(n) \leftarrow \emptyset$ ; /* Liste geeigneter mobiler Endanwender initialisieren */
3   /* Für mobile Endanwender aus MAME  $ME_{DPV}$  berechnen und in  $KMEZL(n)$  einfügen */
4    $eai \leftarrow \emptyset$ 
5   foreach Endanwender  $ea \in MAME$  do
6     if  $EA4(n)$  then
7       if  $ea \in MAME2$  then
8          $ME_{DPV}(ea)[eai] \leftarrow sodpw + ME_{DPV}(ea)[eai]$ ; /* Berücksichtigung des
9         SoD-Constraints */
10      end
11    end
12    switch  $0 < normEZB(ME(ea)) \leq 1 \vee MA_{SAO}(n) = ME_{SAO}(ea)$  do
13      /* Die beste Priorisierung  $ME_{DPV}(ea)$  wird durch eine geometrische
14      Ortsübereinstimmung erreicht */
15      case  $0 < normEZB(ME(ea)) \leq 1$ 
16         $ME_{DPV}(ea)[eai] \leftarrow ME_{DPV}(ea)[eai] + 0 + ME_{DB}(ea) + ME_{RB}(ea) + ME_{ISZ}(ea)$ ;
17         $KMEZL(n) \leftarrow KMEZL(n) \cup \{ea\}$ ;
18      /* Die nächstbeste Priorisierung  $ME_{DPV}(ea)$  wird durch eine symbolische
19      Ortsübereinstimmung erreicht */
20      case  $MA_{SAO}(n) = ME_{SAO}(ea)$ 
21         $ME_{DPV}(ea)[eai] \leftarrow ME_{DPV}(ea)[eai] + 1 + ME_{DB}(ea) + ME_{RB}(ea) + ME_{ISZ}(ea)$ ;
22         $KMEZL(n) \leftarrow KMEZL(n) \cup \{ea\}$ ;
23      /* Dieser Fall deckt die übrigen Fälle ab, in denen keine Ortsübereinstimmung
24      erreicht werden kann */
25      otherwise
26         $ME_{DPV}(ea)[eai] \leftarrow ME_{DPV}(ea)[eai] + 2 + ME_{DB}(ea) + ME_{RB}(ea) + ME_{ISZ}(ea)$ ;
27         $KMEZL(n) \leftarrow KMEZL(n) \cup \{ea\}$ ;
28      end
29    endsw
30     $eai++$ ;
31  end
32  /*  $KMEZL(n), KMEZL(n)[1], \dots, KMEZL(n)[m]$  gemäß  $ME_{DPV}$  aufsteigend sortieren */
33   $tmp \leftarrow \emptyset$ ; /* TMP-Variable für einzusortierenden Wert initialisieren */
34   $tmp2 \leftarrow \emptyset$ ; /* TMP-Variable für einzusortierenden Wert initialisieren */
35   $j \leftarrow 0$ ; /* Hilfszähler initialisieren */
36  for  $i \leftarrow 1$  to  $|KMEZL(n)|$  do
37     $tmp \leftarrow KMEZL(n)[i]$ ;
38     $tmp2 \leftarrow ME_{DPV}(ea)[i]$ ;
39     $j \leftarrow i$ ;
40    while  $(j > 1 \wedge KMEZL(n)[j-1] > ME_{DPV}(ea)[i])$  do
41       $KMEZL(n)[j] \leftarrow KMEZL(n)[j-1]$ ;
42       $ME_{DPV}(ea)[j] \leftarrow ME_{DPV}(ea)[j-1]$ ;
43       $j \leftarrow j-1$ ;
44       $KMEZL(n)[j] \leftarrow tmp$ ;
45       $ME_{DPV}(ea)[j] \leftarrow tmp2$ ;
46    end
47  end
48 end

```

Algorithmus 10.3 : Bestimmung von $KMEZL(n)$ bei Delegationen mit **BoD** Berücksichtigung

Data : Relevante Parameterwerte des mobilen Kontexts (vgl. Tabelle 7.1)

Regeln des mobilen Kontexts (vgl. Tabelle 8.2)

$MAME \subseteq PEA(n)$: Menge angemeldeter und berechtigter mobiler Endanwender

$MASAO(m)$: Symbolischer Ausführungsort der zweiten mobilen Aktivität des **BoD**-Constraints

Prozess-Constraints $EA3$ - $EA5$ der mobilen Aktivität n

$bodpw$: Dieser Wert drückt die spezifische Priorisierung aus

Result : $KMEZL(n)$: Liste geeigneter mobiler Endanwender zur Erzeugung der mobilen Aktivität n

```

1 begin
2    $KMEZL(n) \leftarrow \emptyset$ ; /* Liste geeigneter mobiler Endanwender initialisieren */
3   if  $|MAME| > MASWME(n)$  then
4      $KMEZLH \leftarrow \emptyset$ ; /* Hilfsliste */
5      $EW \leftarrow \emptyset$ ; /* Liste für Eignungswert mobiler Endanwender initialisieren */
6      $eai \leftarrow 0$ 
7     foreach Endanwender  $ea \in MAME$  do
8       if  $EA3(n) \wedge (MESAO(ea) = MASAO(m))$  then
9          $EW[eai] \leftarrow bodpw + SMGBLS(ea) + \frac{1}{MEISZ(ea)} + \frac{1}{MERB(ea)} + MEDB(ea)$ ;
10      end
11      else if  $EA3(n)$  then
12         $EW[eai] \leftarrow SMGBLS(ea) + \frac{1}{MEISZ(ea)} + \frac{1}{MERB(ea)} + MEDB(ea)$ ;
13      end
14       $KMEZLH \leftarrow KMEZLH \cup \{ea\}$ ;
15       $eai++$ ;
16    end
17    /*  $KMEZLH, KMEZLH[1], \dots, KMEZLH[m]$  gemäß  $EW$  absteigend sortieren */
18     $tmp \leftarrow \emptyset$ ; /* TMP-Variable für einzusortierenden Wert initialisieren */
19     $tmp2 \leftarrow \emptyset$ ; /* TMP-Variable für einzusortierenden Wert initialisieren */
20     $j \leftarrow 0$ ; /* Hilfszähler initialisieren */
21    for  $i \leftarrow 1$  to  $|KMEZLH|$  do
22       $tmp \leftarrow KMEZLH[i]$ ;
23       $tmp2 \leftarrow EW[i]$ ;
24       $j \leftarrow i$ ;
25      while  $(j > 1 \wedge KMEZLH[j-1] < EW[i])$  do
26         $KMEZLH[j] \leftarrow KMEZLH[j-1]$ ;
27         $EW[j] \leftarrow EW[j-1]$ ;
28         $j \leftarrow j-1$ ;
29         $KMEZLH[j] \leftarrow tmp$ ;
30         $EW[j] \leftarrow tmp2$ ;
31      end
32    end
33  else
34    foreach Endanwender  $ea \in MAME$  do
35       $KMEZL(n) = KMEZL(n) \cup \{ea\}$ ;
36    end
37  end
38 end

```

<i>A</i>	Anpassung	A1	A2	A3
SoD-SoD-Constraint-Kombination (A1,A2) u. (A2,A3)				
9	Das Backup berücksichtigt nicht nur A2, sondern auch A3.	✓	✓	✓
BoD-BoD-Constraint-Kombination (A1,A2) u. (A2,A3)				
10	Algorithmus 10.3 wird so verändert, dass nicht nur der symbolische Ausführungsort von A2, sondern auch derjenige von A3 berücksichtigt wird.	✓	X	X
11	Die Anpassungen für A2 werden auch für A3 angewandt.	X	✓	✓
12	Die Anpassungen für A2 werden auch für A3 angewandt.	X	✓	✓
SoD-BoD-Constraint-Kombination (A1,A2) u. (A2,A3)				
8	Algorithmus 10.2 wird so verändert, dass die Delegationslistenanpassung nicht nur für A1, sondern auch für A2 erfolgt (vgl. Abb. 10.7).	✓	✓	X
BoD-SoD-Constraint-Kombination (A1,A2) u. (A2,A3)				
8	Algorithmus 10.2 wird bereits für A1 angewandt und nicht erst für A2, der ersten Aktivität des SoD -Constraints. Praktisch gesehen handelt es sich um einen vorgezogenen SoD -Constraint zu A1.	✓	X	X
Card-BoD-Constraint-Kombination (A1,A2) u. (A2,A3)				
13	Der mobile Endanwender von A2 ergibt sich aus der Ausnahmebehandlung von A1 (vgl. Anpassung 13).	✓	✓	–
Card-SoD-Constraint-Kombination (A1,A2)				
13	Algorithmus 10.2 wird auf alle Fälle für A1 angewandt, da nicht nur der Card -Constraint, sondern auch der SoD -Constraint gilt.	✓	–	–
BoD-Card-Constraint-Kombination (A1,A2)				
Die BoD-Card-Constraint-Kombination wird wie eine BoD-BoD-Constraint-Kombination behandelt.				
SoD-Card-Constraint-Kombination (A1,A2)				
9	Das Backup berücksichtigt A2 wie bei einer SoD-SoD-Constraint-Kombination.	✓	✓	–
✓=trifft zu, X=trifft nicht zu, –=irrelevant, A=Anpassung				

Tabelle 10.7: Anpassungen für kombinierte Prozess-Constraints bei der Ausnahmebehandlung

Anforderung	Maßnahmen
Funktionale Anforderungen mobiler Aktivitäten	
Anforderung 8	Prozess-Constraints werden explizit unterstützt und berücksichtigen ferner den mobilen Kontext.
Funktionale Anforderungen des mobilen Kontexts	
Vergleiche zu den Anforderungen 10-15 (1) die Tabellen 10.4, 10.5, 10.6 und 10.7 sowie (2) die Algorithmen 10.1, 10.2, und 10.3	
Anforderung 10	Prozess-Constraints berücksichtigen den Ausführungsort.
Anforderung 11	Prozess-Constraints berücksichtigen den Netztyp.
Anforderung 12	Prozess-Constraints berücksichtigen den Formfaktor.
Anforderung 13	Prozess-Constraints berücksichtigen den Energiestatus.
Anforderung 14	Prozess-Constraints berücksichtigen die Dringlichkeit.
Anforderung 15	Prozess-Constraints berücksichtigen das Endanwenderverhalten.
Nicht-funktionale Anforderungen mobiler Aktivitäten	
Anforderung 16	Prozess-Constraints werden explizit mit den Ausführungs- und Ausnahmebehandlungskonzepten verknüpft.
Anforderung 17	Prozess-Constraints erfordern bis auf die Rückmeldung zu Anfragen eines Systemwechsels keine manuellen Entscheidungen.
Anforderung 18	Prozess-Constraints berücksichtigen frühzeitig Laufzeitprobleme (s. z.B. Abschnitt 10.4.1)

Tabelle 10.8: Beitrag der Prozess-Constraints zur robusten Ausführung mobiler Aktivitäten

Man verdirbt einen Jüngling am sichersten, wenn man ihn verleitet, den Gleichdenkenden höher zu achten als den Andersdenkenden.

Friedrich Nietzsche (1844-1900)

11

Architekturaspekte

Gegenstand dieses Kapitels bildet die technische Realisierung und Machbarkeit der in den vorangehenden Kapiteln vorgestellten Konzepte. Hierzu sind zwei Schritte notwendig: Erstens muss eine geeignete Architektur definiert werden [Haa13, Kre14, PLRH12, PMLR14, SSP⁺14, PMR14], zweitens ist eine geeignete Informationsarchitektur [IAT14, PLRH12, PMLR14] festzulegen. Dieses Kapitel behandelt beide Aspekte und diskutiert weitere Implementierungsdetails bezogen auf die Integration mobiler Aktivitäten in eine Prozessumgebung.

Kapitel 11 gliedert sich wie folgt: Abschnitt 11.1 diskutiert grundsätzliche Möglichkeiten im Hinblick auf die Realisierung einer geeigneten Architektur für mobile Aktivitäten. Abschnitt 11.2 stellt die Komponenten des Prozess-Klienten vor, während Abschnitt 11.3 die Komponenten der Ausführungsebene behandelt. Abschnitt 11.4 stellt die Modellierungsumgebung sowie deren Komponenten und Abschnitt 11.5 die Informationsarchitektur des mobilen Prozess-Klienten vor. Abschnitt 11.6 diskutiert verwandte Ansätze und Abschnitt 11.7 fasst das Kapitel zusammen.

11.1 Einleitung

Der Entwurf einer geeigneten Architektur für die Integration mobiler Aktivitäten in eine Prozessumgebung erfolgt durch Erweiterung einer bestehenden Architektur. Da viele etablierte Konzepte existieren, verspricht dieser Ansatz die Wiederverwendung existierender Architekturkomponenten. Grundsätzlich existieren drei Ansätze, um eine solche Integration zu realisieren (vgl. Abb. 11.1①). Ihre Unterschiede fasst Tabelle 11.1 zusammen.

In dieser Arbeit wird Variante 3 verwendet, um mobile Aktivitäten technisch zu realisieren. Variante 3 bedeutet, dass eine separate Komponente entwickelt wird, die ausschließlich für die Ausführung mobiler Aktivitäten verantwortlich ist. Daher muss die Komponente mit einem existierenden *PrMS* integriert werden. Obwohl die Integration Zusatzaufwand bedeutet, ist die prototypische Realisierung der vorgestellten Konzepte so effektiv möglich.

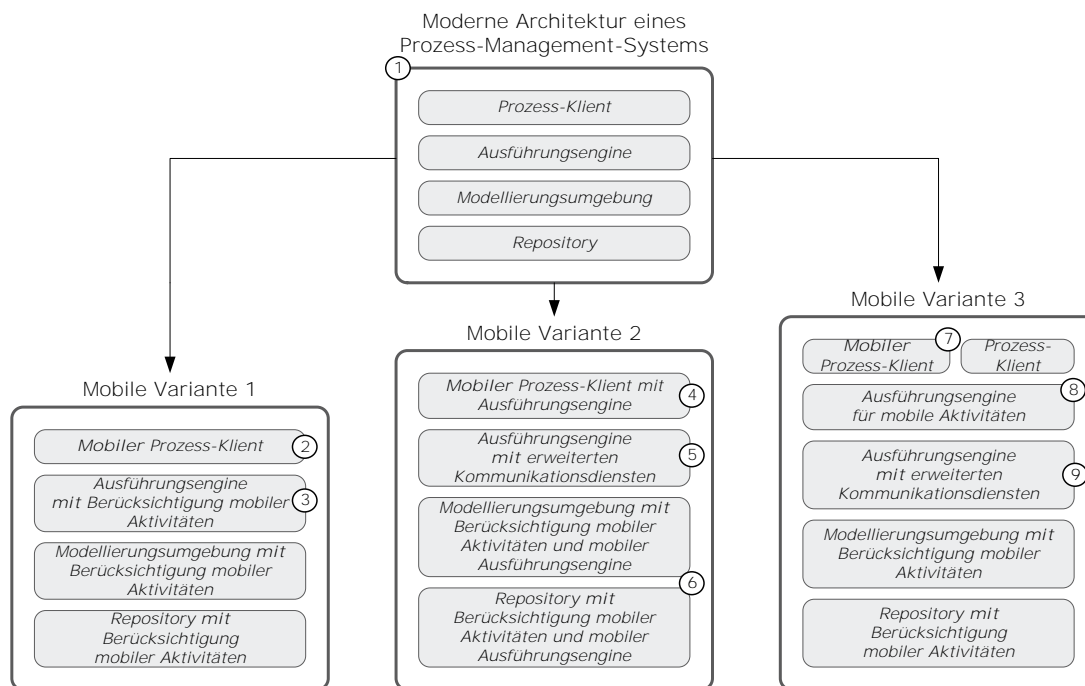


Abbildung 11.1: Varianten für eine mobile Prozess-Management-Architektur

Variante	Erläuterung
1	Bei Variante 1 werden die bestehenden Komponenten des Prozess-Klienten ② und der Ausführungseingine ③ erweitert, sodass mobile Aktivitäten integriert und damit auch ausgeführt werden können. Ferner müssen die Modellierungsumgebung und das Repository angepasst werden.
2	Bei Variante 2 ist ein Prozess-Klient nicht nur in der Lage, einzelne mobile Aktivitäten auszuführen, sondern er besitzt auch eine eigene Ausführungseingine ④. Diese ist in der Lage, Prozesse oder Fragmente eines Prozesses in autonomer Art und Weise auf einem Smart-Mobilgerät auszuführen. Eine solche Ausführungseingine für Smart-Mobilgeräte zu entwickeln, stellt eine komplexe Aufgabe dar [PTKR10, PTR10, SSP ⁺ 14]. Ferner muss dazu die bisherige Ausführungseingine ⑤ angepasst werden. Schließlich müssen die Modellierungskomponente und das Repository ⑥ ebenfalls angepasst werden.
3	Variante 3 stellt eine Abwandlung von Variante 1 dar. Die Funktionalität ist identisch, nur dass für die mobile Aktivitätsausführung separate Komponenten entwickelt werden ⑦+⑧. Diese werden über geeignete Schnittstellen mit der existierenden Komponente zur Aktivitätsausführung ⑨ verbunden. Da diese Variante die Möglichkeit bietet, für die neuen Komponenten ⑦+⑧ bei Bedarf die verwendeten Technologien zu wechseln (weil diese beispielsweise auf mobile Systeme speziell Rücksicht nehmen), ist sie im Vergleich zu Variante 1 potentiell einfacher zu realisieren. Auf der anderen Seite müssen mehrere Konzepte redundant entwickelt werden und Änderungen müssen mehrfach gepflegt werden. Außerdem ist es wichtig, geeignete Schnittstellen und Protokolle zu wählen, um die Komponenten zu verbinden.

Tabelle 11.1: Erläuterungen zu den Varianten einer mobilen Prozess-Management-Architektur

11.2 Mobiler Prozess-Klient

Dieser Abschnitt stellt die Architekturkomponenten für den mobilen Prozess-Klienten vor. Diese sind in Abbildung 11.2 dargestellt.

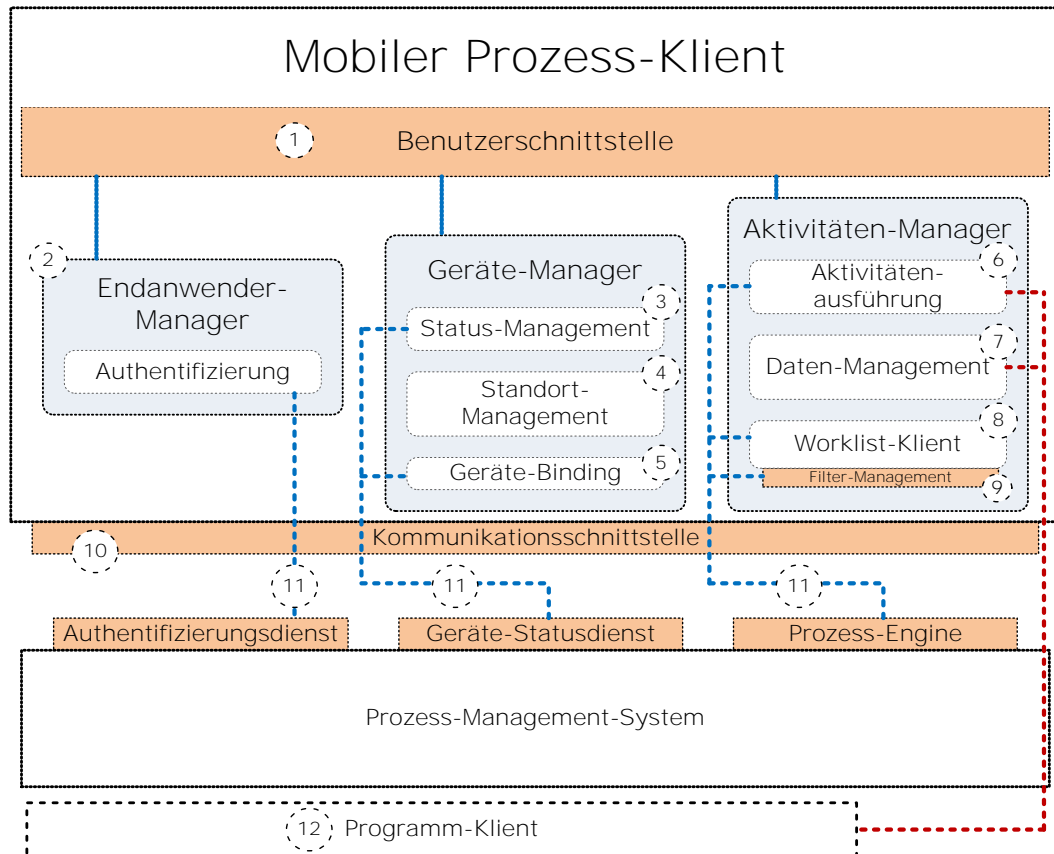


Abbildung 11.2: Architekturkomponenten mobiler Prozess-Klient

11.2.1 Benutzerschnittstelle

Die Benutzerschnittstelle (vgl. Abb. 11.2①) ist die Schnittstelle zum mobilen Endanwender. Sie zeigt dem Endanwender die Worklist mit Aktivitäteneinträgen an, ermöglicht eine Authentifizierung gegenüber dem *PrMS* und steuert die Ausführung einzelner Aktivitäten. Für die Ausführung von mobilen Aktivitäten wird einerseits die automatische Generierung der Benutzerschnittstelle¹, andererseits der externe Aufruf von Anwendungen auf dem Smart-Mobilgerät unterstützt.

¹Weitere Details diskutieren [KHR12, SSPR15].

11.2.2 Endanwender-Manager

Der Endanwender-Manager (vgl. Abb. 11.2②) besteht aus einem Modul, das sich um die Authentifizierung von Endanwendern gegenüber dem *PrMS* kümmert. Dazu kann sich der Endanwender mit seinen *Credentials* [CRE14] am mobilen Prozess-Klienten anmelden. Das Authentifizierungs-Modul des Endanwender-Managers überträgt die Daten sodann zum *PrMS*, woraufhin es eine Rückmeldung erhält, welche Rollen der Endanwender mit den eingegebenen *Credentials* einnehmen darf. Er wählt schließlich eine Rolle aus und kann nun Aktivitäten, für die er entsprechende Rechte besitzt, am mobilen Prozess-Klienten ausführen.

11.2.3 Geräte-Manager

Der Geräte-Manager besteht aus drei Modulen: Modul 1 (vgl. Abb. 11.2③) ist für das Status-Management des Smart-Mobilgeräts zuständig. So muss erkannt werden, wann ein Smart-Mobilgerät offline ist. Der hierzu angewandte Mechanismus ist die regelmäßige Übersendung einer *alive*-Nachricht des Smart-Mobilgeräts an das *PrMS*. Das Status-Management Modul sendet diese Nachricht in konfigurierbaren Abständen an das *PrMS*.

Modul 2 (vgl. Abb. 11.2④) bestimmt die Lokation des Smart-Mobilgeräts und sendet diese an das *PrMS*. Es kann sich um eine GPS-Bestimmung oder WLAN-Ortsbestimmung handeln.

Modul 3 (vgl. Abb. 11.2⑤) ist für das Geräte-Binding (vgl. Listing 11.1) verantwortlich. Ausschließlich autorisierte Smart-Mobilgeräte sollen Aktivitäten ausführen können. Dazu sendet ein Prozess-Klient auf einem Smart-Mobilgerät nach Aufruf seine MAC-Adresse an das *PrMS*. Ist diese (bzw. das Smart-Mobilgerät mit dieser MAC-Adresse) zur Ausführung von Aktivitäten autorisiert, wird ein Sicherheits-Token ausgetauscht. Dieses wird bei der gesamten Kommunikation mitgesandt, um sicherzustellen, dass Aktivitäten nur von autorisierten Smart-Mobilgeräten ausgeführt werden.

11.2.4 Aktivitäten-Manager

Der Aktivitäten-Manager umfasst ebenfalls drei Module: Modul 1 (vgl. Abb. 11.2⑥) ist für die eigentliche Aktivitätsausführung zuständig. Letztere ruft externe Anwendungen auf, regelt den Datenaustausch mit dem *PrMS* und versorgt die Komponente für die Benutzerschnittstelle mit Informationen zu den Aktivitäten. Modul 2 (vgl. Abb. 11.2⑦) kommuniziert mit der Aktivitätsausführung und bewerkstelligt die Zwischenspeicherung aller lokal erzeugten Daten. Modul 3 (vgl. Abb. 11.2⑧) ist schließlich für die Worklist zuständig. Es verwaltet die Worklist-Einträge (inkl. ihrer Statusinformationen).

Ein wichtiger Aspekt ist das lokale Daten-Management. Letzteres bedeutet, dass auf dem Smart-Mobilgerät erzeugte Daten dort zwischengespeichert werden, da ansonsten Funktionen wie Offline-Bearbeitung und Delegation nicht möglich sind. Generell arbeiten die Aktivitätsausführung und das lokale Daten-Management nach folgendem Prinzip: Bei der Aktivitätsausführung erzeugte Daten werden stets lokal zwischengespeichert und mit der *ID* der aktuell ausgeführten Aktivität verknüpft. In Bezug auf zwischengespeicherte Daten müssen zwei Datenarten unterschieden werden.

1. Es gibt Daten, die unter der vollen Kontrolle der Aktivitätsausführung stehen. Dies ist der Fall, wenn die Aktivitätsausführung ausschließlich im Aktivitäten-Manager stattfin-

det, d.h. es muss kein weiteres Programm zur Ausführung gerufen werden (z.B. eine lokale Kalender-App). Andererseits ist dies der Fall, wenn die API des zu rufenden Programms den Datenbestand stets zum Abruf bereithält.

2. Es gibt weiters Daten, die nicht unter der vollen Kontrolle der Aktivitätsausführung stehen. In diesem Fall stellt ein zu rufendes Programm die Daten erst nach seiner Beendigung zur Verfügung. D.h. es existiert keine spezielle API, welche die Daten in kontinuierlicher Art und Weise zur Verfügung stellt.

Das lokale Daten-Management und die Aktivitätsausführung berücksichtigen beide Arten von Daten. Das Daten-Management als eigenständige Komponente zu konzipieren ist schon deshalb wichtig, um später entsprechende Schnittstellen für die technische Realisierung spezifizieren zu können.

Das Daten-Management und die Aktivitätsausführung sind die Komponenten, die zusammen den Programm-Klienten (vgl. Abb. 11.2⑫) bilden.

11.2.5 Weitere Komponenten

Schließlich zeigen die Markierungen ⑨ und ⑩ in Abbildung 11.2 die Kommunikationsschnittstellen zwischen mobilen Prozess-Klienten und *PrMS*. Für jeden Manager des mobilen Prozess-Klienten wird eine solche Schnittstelle seitens des *PrMS* benötigt.

11.3 Ausführungsengine

Dieser Abschnitt diskutiert die Architekturkomponenten, die für das *PrMS* verändert und ergänzt werden müssen, um mobile Aktivitäten ausführen zu können (vgl. Abb. 11.3). Dabei werden die neu hinzukommenden Komponenten und die zu verändernden Komponenten farblich hervorgehoben (vgl. Abb. 11.3).

11.3.1 Endanwender-Manager

Der Endanwender-Manager (vgl. Abb. 11.3①) erfüllt zwei Aufgaben. Erstens behandelt er Authentifizierungsanfragen des mobilen Prozess-Klienten (Authentifizierungs-Handler). Diese werden an das Repository weitergeleitet, wo erfragt wird, ob der mobile Endanwender im Repository existiert und seine Zugangsdaten (Credentials) richtig eingegeben hat (vgl. Markierung ⑨ in Abb. 11.3). Zweitens bindet der Endanwender-Manager einen mobilen Endwender, der durch den Authentifizierungs-Handler bestätigt wurde, an eine Session (Session Handler). Wie diese Bindung konkret aussieht, zeigt Abbildung 11.4. Konkret wird ein *Shared Secret* erstellt, das dazu verwendet wird, den entsprechenden mobilen Endanwender im System zu verwalten.

11.3.2 Geräte-Manager

Der Geräte-Manager (vgl. Abb. 11.3②) ist das Gegenstück eines mobilen Prozess-Klienten auf Seiten des *PrMS*. Er überwacht den Gerätestatus (Geräte-Status Handler) (vgl. Abb. 11.3③) und bindet Smart-Mobilgeräte an die Ausführungsengine (Geräte-Binding Handler).

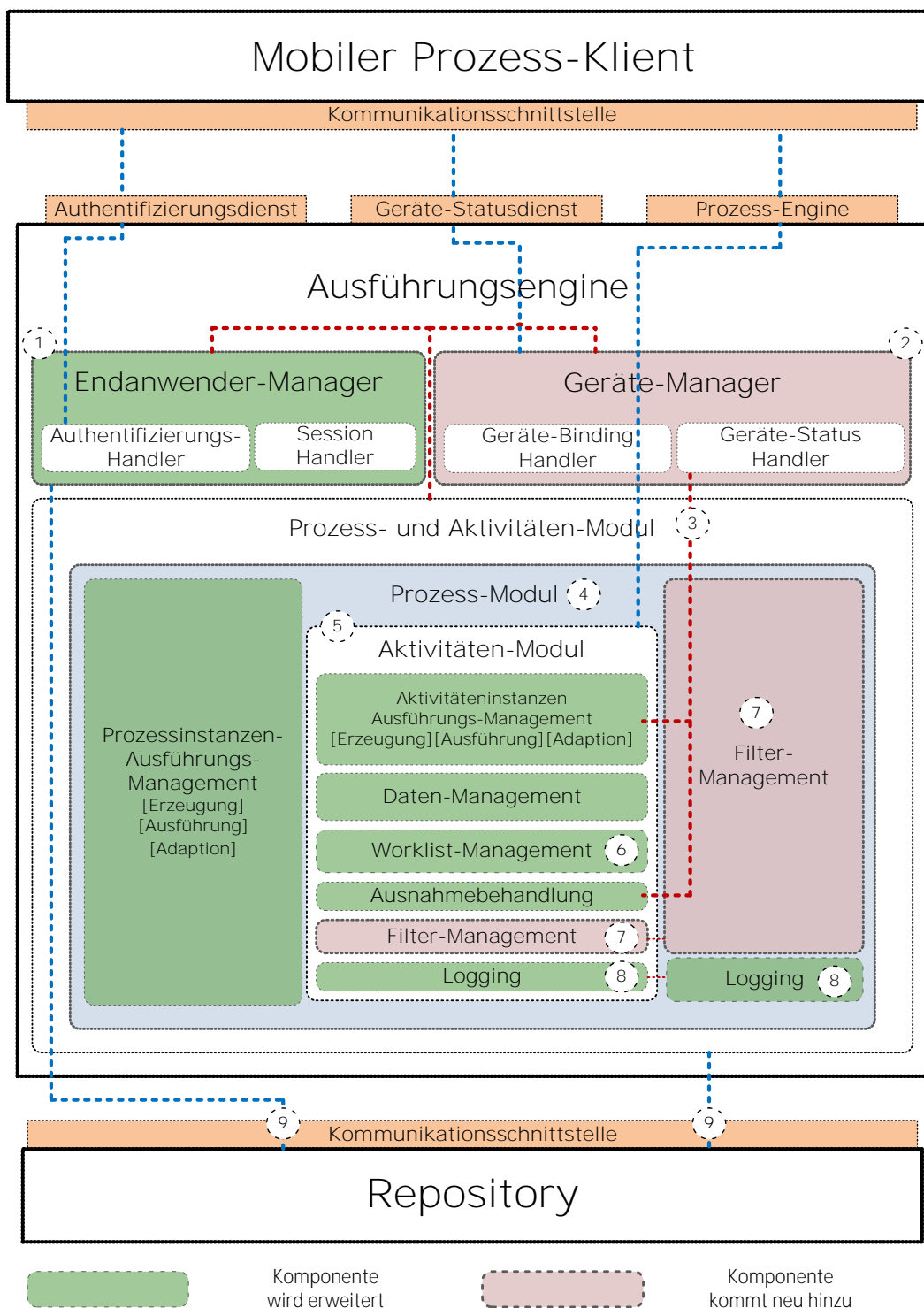


Abbildung 11.3: Architekturkomponenten einer mobilen Aktivitätenausführung

Listing 11.1 zeigt, wie das Geräte-Binding durchgeführt wird. Ein wichtiger Aspekt des Geräte-Managers ist der Status-Handler, der direkt mit der Ausführungseengine (vgl. Abb. 11.3⑤) kommuniziert. So kann in Abhängigkeit des Geräte-Status entschieden werden, ob entweder das Smart-Mobilgerät zur Ausführung einer mobilen Aktivität geeignet ist (z.B. wegen eines guten Batterieladestatus) oder eventuell eine Ausnahmebehandlung (z.B. wenn das Smart-Mobilgerät nicht verfügbar ist) gestartet werden muss. Für jedes Smart-Mobilgerät läuft ein Thread zur Überwachung des Geräte-Status, der wiederum für die Alive-Messages des Smart-Mobilgeräts verantwortlich ist.

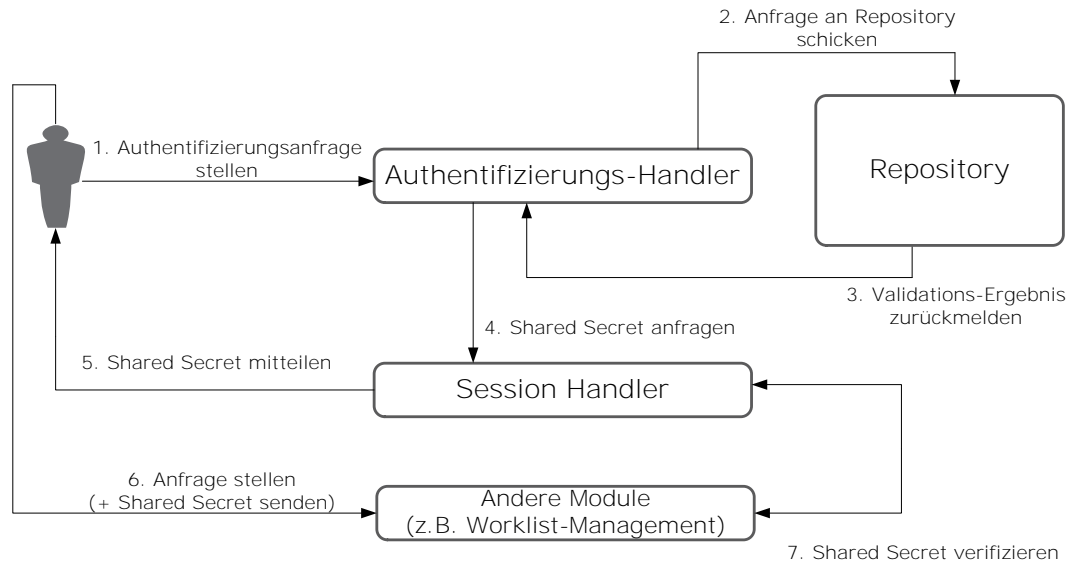


Abbildung 11.4: Session-Handling für mobile Endanwender

11.3.3 Ausführungseengine

Für die Ausführungseengine (vgl. Abb. 11.3③) müssen Komponenten ergänzt und erweitert werden, welche die Ausführung mobiler Aktivitäten ermöglichen. Anzumerken ist, dass die vier Komponenten aus der Referenzarchitektur (d.h. Erzeugung, Ausführung, Adaption und Logging; vgl. Kapitel 6) weiter verfeinert werden. Es wird unterschieden, ob es sich um Komponenten, die einzelne Aktivitäten ausführen (Aktivitäten-Modul), oder Komponenten, die für komplette Prozessinstanzen (Prozess-Modul) verantwortlich sind, handelt.

Listing 11.1: pairDevice

```

1  /**
2   * Pair a new device
3   *
4   * @param macAddress Mac Address if the device to be paired
5   * @return paired DeviceModel instance
6   */
7  public DeviceModel pairDevice(String macAddress)
8      throws DeviceAlreadyPairedException {
9      if (!mDeviceMacAddressMap.containsKey(macAddress)) {
10
11          DeviceModel device = new DeviceModel(macAddress);

```

```

12     String token ;
13
14     do {
15         token = TokenFactory.getToken();
16     } while (mDeviceTokenMap.containsKey(token));
17
18     device.setPairingToken(token);
19     device.save();
20     mDeviceTokenMap.put(device.getPairingToken(), device);
21     mDeviceMacAddressMap.put(device.getMacAddress(), device);
22     return device;
23
24 }
25 else { throw new DeviceAlreadyPairedException(); }
26 }

```

Im Folgenden werden die Komponenten der Ausführungseengine im Detail vorgestellt: Das Prozess-Modul (vgl. Abb. 11.3④) führt Prozessinstanzen aus und interagiert mit dem Aktivitäten-Modul, um einzelne Aktivitäten einer Prozessinstanz auszuführen. Da sich das Filter-Management auch auf Prozessinstanzen auswirkt, wird diese Komponente hinzugefügt und mit der bestehenden Komponente zum Ausführungs-Management von Instanzen verbunden. Da mobile Aktivitäten und Prozessinstanzen auch Einträge für das Logging erzeugen, muss die Logging-Komponente (vgl. Abb. 11.3⑧) ebenfalls erweitert werden.

Das Aktivitäten-Modul (vgl. Abb. 11.3⑤) ist für die Ausführung einzelner Aktivitäten zuständig. Alle Einzelkomponenten (vgl. z.B. Abb. 11.3⑥) müssen zur Ausführung mobiler Aktivitäten ergänzt werden (vgl. Abb. 11.3⑤). Weiters muss das Filter-Management als Komponente hinzugefügt werden. Das Filter-Management (vgl. Abb. 11.3⑥&⑦) wiederum wird sowohl auf Aktivitäten- als auch Prozessinstanz-Ebene angewandt. In diesem Kontext soll in Erinnerung gerufen werden, dass bei der Erzeugung und Ausführung mobiler Aktivitäten möglichst nur die geeignetsten mobilen Endanwender ausgewählt werden sollen. Zu diesem Zweck wurden in Kapitel 8 Kriterien zur kontextbezogenen Zuteilung vorgestellt. Das Filter-Management setzt diese kontextbezogene Zuteilung nun um. Dazu existiert für jedes Kriterium (z.B. Ortseignung) ein Filter, der auf die Liste der potentiellen Endanwender einer betrachteten mobilen Aktivität angewandt wird.

Da einige Kriterien (z.B. GP_{ISZ})² Einschränkungen auf Prozess-Typ-Ebene mit sich bringen, ist das Filter-Management nicht nur auf Aktivitäten-Ebene vorhanden (vgl. Abb. 11.3⑤), sondern auch auf Instanz-Ebene (vgl. Abb. 11.3④). Listing 11.2 zeigt beispielhaft das Anwenden eines *Lokationsfilters*.

Der Endanwender-Manager und die Ausführungseengine kommunizieren über Schnittstellen mit dem Repository (vgl. Abb. 11.3⑨). Der Endanwender-Manager gibt über die Schnittstelle Authentifizierungsanfragen an das Repository weiter. Die Ausführungseengine nutzt die Schnittstelle, um alle dauerhaft zu speichernden Daten in das Repository zu übertragen. Ferner werden alle bereits gespeicherten Daten von dort abgerufen.

Listing 11.2: LocationMatchingFilter

```

1  /**
2   * Filter to calculate the normalized location factor (nlf)
3   * for all users of a task *
4   */
5  public class LocationMatchingFilter implements IBaseFilter {
6
7      private MobileUserManager mMobileUserManager;
8      private DeviceActivityManager mDeviceActivityManager;
9      private boolean isInit = false;
10     private Map<Integer, Float> mLocationFactors;

```

²Instant-Shut-down Referenzwert

```

11
12 public LocationMatchingFilter() {
13     mLocationFactors = new HashMap<>();
14     this.mMobileUserManager = MobileUserManager.getInstance();
15     this.mDeviceActivityManager = DeviceActivityManager.getInstance();
16 }
17
18 @Override
19 public void apply(MobileTask mobileTask) {
20     if (!isInit) {
21         // Get Task Location
22         TaskLocation taskLocation = mobileTask.getLocation();
23         boolean hasLocation = taskLocation != null;
24         List<Integer> userList = mobileTask.getUserList();
25
26         for (int userId : userList) {
27             Float nlf = -1f;
28             DeviceLocation deviceLoc;
29             try {
30                 // get current user Position
31                 MobileUserModel currentUser = mMobileUserManager.getMobileUserById(userId);
32                 String deviceToken = mDeviceActivityManager.getBoundDeviceToken(currentUser);
33                 DeviceAliveMessage dam = mDeviceActivityManager.getDeviceAliveMessage(deviceToken);
34                 deviceLoc = dam.getLocation();
35             } catch (NoSuchUserException | NoDeviceAliveMessageException e) { deviceLoc = null; }
36
37             // 1. check if in catchment area .. if not, set nlf = -1
38             if (deviceLoc != null && taskLocation.isInCatchmentArea(deviceLoc)) {
39                 // 2. if in catchment area calculate nlf
40                 Float distance = taskLocation.getDistanceTo(deviceLoc);
41                 if (distance != 0) {
42                     nlf = taskLocation.getCatchmentAreaRadius() / distance;
43                 } else { nlf = 1f; }
44             }
45
46             mLocationFactors.put(userId, nlf);
47         }
48         isInit = true;
49     }
50 }

```

11.4 Prozess-Modellierung

Dieser Abschnitt diskutiert die Änderungen, die für eine Modellierungsumgebung durchgeführt werden müssen, um mobile Aktivitäten geeignet zu integrieren. Abbildung 11.5 zeigt die Änderungen im Überblick. Konkret muss die Oberfläche einer Modellierungsumgebung (vgl. Abb. 11.5①) um die Funktionen erweitert werden (vgl. Abb. 11.5②), die das Modul für mobile Aktivitäten zur Verfügung stellt (vgl. Abb. 11.5③). Ferner stellt das Modul für mobile Aktivitäten über die Schnittstelle zum Repository (vgl. Abb. 11.5④) die Abhängigkeitsüberprüfung (vgl. Abb. 11.5④) und das Filter-Management (vgl. Abb. 11.5⑤) während der Modellierung zur Verfügung.

Wie in Kapitel 8 vorgestellt, sind bereits während der Modellierung Berechnungen (vgl. Abschnitt 8.2) notwendig, die von diesen beiden Komponenten bewerkstelligt werden. Die in Abbildung 11.5② gezeigte Komponente ermöglicht die Abfrage dieser Ergebnisse durch einen Prozessmodellierer.

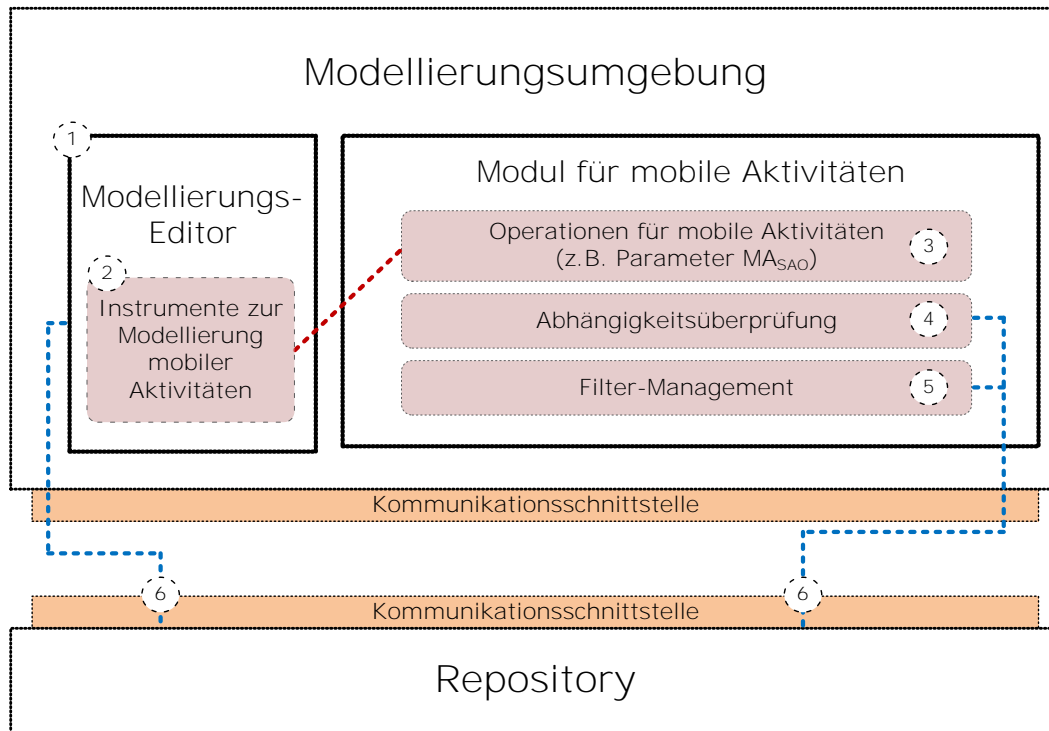


Abbildung 11.5: Architekturkomponenten für Prozess-Modellierung

11.5 Informationsarchitektur des mobilen Prozess-Klienten

Dieser Abschnitt stellt die Informationsarchitektur³ [IAT14] des mobilen Prozess-Klienten vor. Eine für den Benutzer verständliche Informationsarchitektur ist besonders im mobilen Arbeitsumfeld wichtig, da ein Smart-Mobilgerät deutlich weniger Anzeigefläche als ein Desktop-System bietet. Diesem Umstand muss Rechnung getragen werden, um mobile Aktivitäten sinnvoll zu integrieren (vgl. Abb. 11.6). Weitere Aspekte diskutiert [PLRH12, PMLR14].

Ein mobiler Endanwender loggt sich zunächst am mobilen Prozess-Klienten ein (vgl. Abb. 11.6①). Dabei wird der in Abbildung 11.4 dargestellte Ablauf im Hintergrund durchgeführt. Nach dem Einloggen sieht der mobile Endanwender seine Aktivitäten (vgl. Abb. 11.6②). Hier-von ausgehend hat er fünf Auswahlmöglichkeiten:

1. Er kann die Liste offener Aktivitäten betrachten.
2. Er kann er sich die Delegationsanfragen betrachten (vgl. Abb. 11.6④). Handelt es sich speziell um Anfragen an ihn, wird er automatisch in diese Ansicht geleitet. Dies erfolgt nur dann, wenn der mobile Endanwender aktuell keine mobile Aktivität ausführt (vgl. Abb. 11.6⑦).
3. Er kann Aktivitäten starten (vgl. Abb. 11.6⑦).
4. Er kann Aktivitäten annehmen (vgl. Abb. 11.6③).

³Die Informationsarchitektur definiert, wie die Benutzerschnittstelle aufgebaut ist. D.h. es wird festgelegt, aus welchen Fenstern eine Anwendung besteht und wie diese untereinander verknüpft sind.

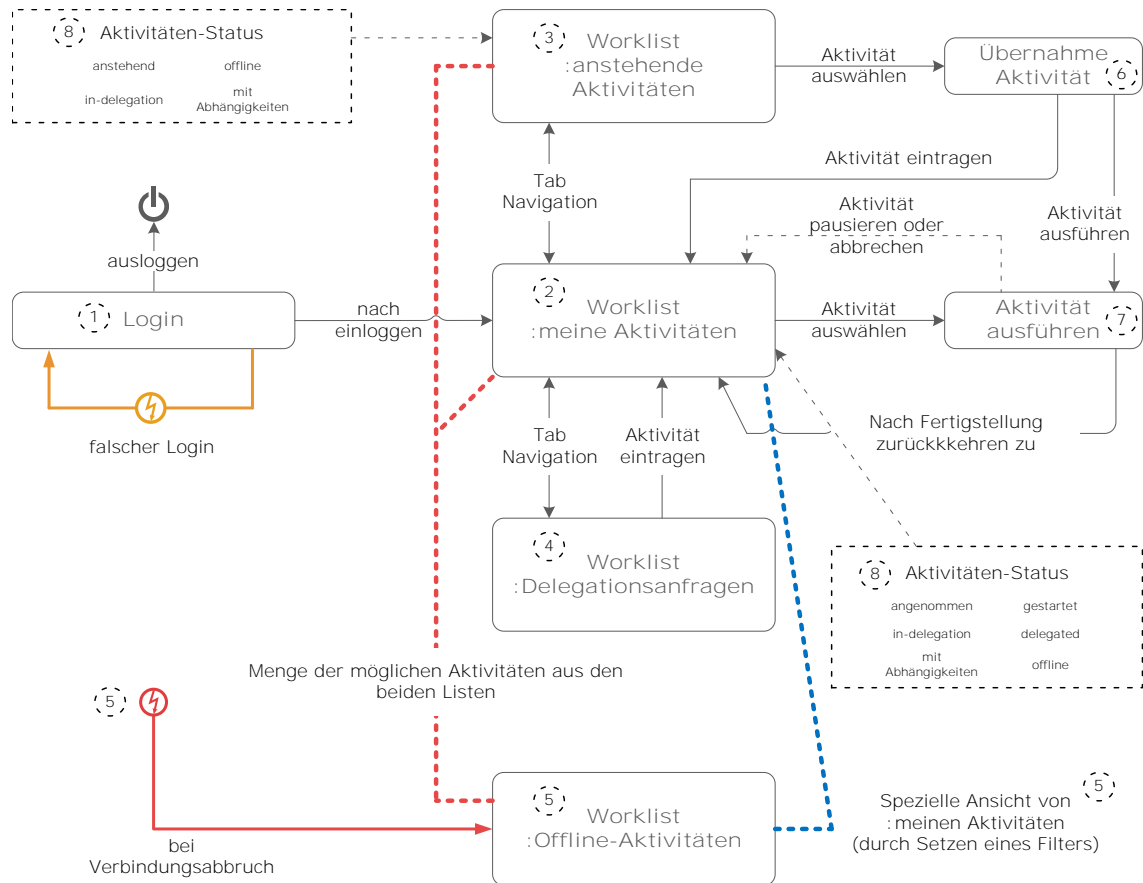


Abbildung 11.6: Informationsarchitektur mobiler Prozess-Klient

5. Er kann sich alle Offline-Aktivitäten anzeigen lassen (vgl. Abb. 11.6⑤).

Schließlich sind in Abbildung 11.6 die Statusinformationen ⑧ zu sehen, die in den Listen der angenommenen Aktivitäten (vgl. Abb. 11.6②) und offenen Aktivitäten (vgl. Abb. 11.6③) verwaltet werden.

11.6 Verwandte Ansätze

Prozess-Management-Architekturen, die mobile Aktivitäten berücksichtigen⁴, werden in verwandten Ansätzen bislang kaum diskutiert. Dass Prozess-Architekturen fundamental sind, diskutiert [Kre14]. [PMLR14] zeigt ferner auf, dass der mobile Prozess-Klient aus Sicht der Benutzerakzeptanz fundamental ist.

Ein Architektur-zentriertes Projekt ist DEMAC [DEM14]. Hier werden ebenfalls Architektur-betrachtungen angestellt [KZL07b, Kun05, ZL10]. Jedoch sind diese nicht so umfangreich wie die vorangehend diskutierten Architektur-aspekte. Da in DEMAC als Grundlage für mobile Prozesse Web Services dienen, stehen Architekturbetrachtungen im Fokus, die sich mit Web

⁴D.h. Architekturen, die insbesondere auch einen mobilen Prozess-Klienten bieten.

Services befassen.

Ein weiteres Projekt, das mobile Prozesse auf Web Service Basis verfolgt, ist SLIVER [HHGR06]. Auch dieses Projekt fokussiert auf Web Services. Zuletzt sei SAMPROC [Sch09] erwähnt. Dieses Projekt betrachtet dynamische Code-Migration zwischen Smart-Mobilgeräten auf Basis von Web Service Techniken [SDGH09, SKHR08]. Auch dieses Projekt stellt Architekturbetrachtungen an, allerdings mit einer anderen Zielsetzung.

11.7 Zusammenfassung

Dieses Kapitel hat gezeigt, wie eine *PrMS*-Referenzarchitektur erweitert bzw. verändert werden muss, um die vorgestellten Konzepte mobiler Aktivitäten zu realisieren. Es wurde gezeigt, dass verschiedene Varianten zur Realisierung einer mobilen Prozess-Management-Architektur existieren. Die gewählte Variante wurde vorgestellt und Herausforderungen diskutiert.

Teil IV

Validation der Konzepte

Von allen Geschenken, die uns das Schicksal gewährt, gibt es kein größeres Gut als die Freundschaft - keinen größeren Reichtum, keine größere Freude.

Epikur (341 - 271 v. Chr.)

12

Prototypische Realisierung

Im Teil III dieser Arbeit wurden fortschrittliche Konzepte vorgestellt, auf deren Grundlage sich mobile Aktivitäten in einer Prozessumgebung robust ausführen lassen. Dieses Kapitel zeigt wie sich diese Konzepte prototypisch realisieren und anwenden lassen.

Kapitel 12 ist wie folgt aufgebaut: In Abschnitt 12.1 wird die Vorgehensweise bei der Entwicklung des Prototyps erläutert. Abschnitt 12.2 stellt dann die wesentlichen Komponenten des Prototyps vor. Die Abschnitte 12.3 und 12.4 beschreiben die realisierten Schnittstellen zum *PrMS* bzw. dem mobilen Prozess-Klienten. In Abschnitt 12.5 wird der Kernel des Prototyps im Detail vorgestellt, während Abschnitt 12.6 das Repository beschreibt. In Abschnitt 12.7 wird der mobile Prozess-Klient skizziert. Abschnitt 12.8 diskutiert weitere Aspekte und Abschnitt 12.9 fasst das Kapitel zusammen.

12.1 Einleitung

Die prototypische Implementierung der entwickelten Konzepte ist eine wichtige Säule, um deren Machbarkeit und Anwendbarkeit zu demonstrieren. In Kapitel 11 wurde bereits die technische Integration der Konzepte in ein bestehendes *PrMS* diskutiert. Die Entwicklung des Prototyps folgte diesem Ansatz (vgl. Abb. 12.1).

Der Prototyp besteht im Kern aus vier Komponenten. Die erste Komponente ist eine Service-orientierte Middleware (vgl. Abb. 12.1①), welche die Konzepte für mobile Aktivitäten realisiert. Die zweite Komponente realisiert den mobilen Prozess-Klienten (vgl. Abb. 12.1②). Dieser kommuniziert mittels REST-Techniken mit der ersten Komponente (vgl. Abb. 12.1③). Die dritte Komponente ist für die Modellierung mobiler Aktivitäten verantwortlich (vgl. Abb. 12.1④), während die vierte Komponente die Schnittstelle zum *PrMS* (vgl. Abb. 12.1⑤) bildet. Abbildung 12.2 veranschaulicht den Integrationsablauf zwischen Prototyp und *PrMS*.

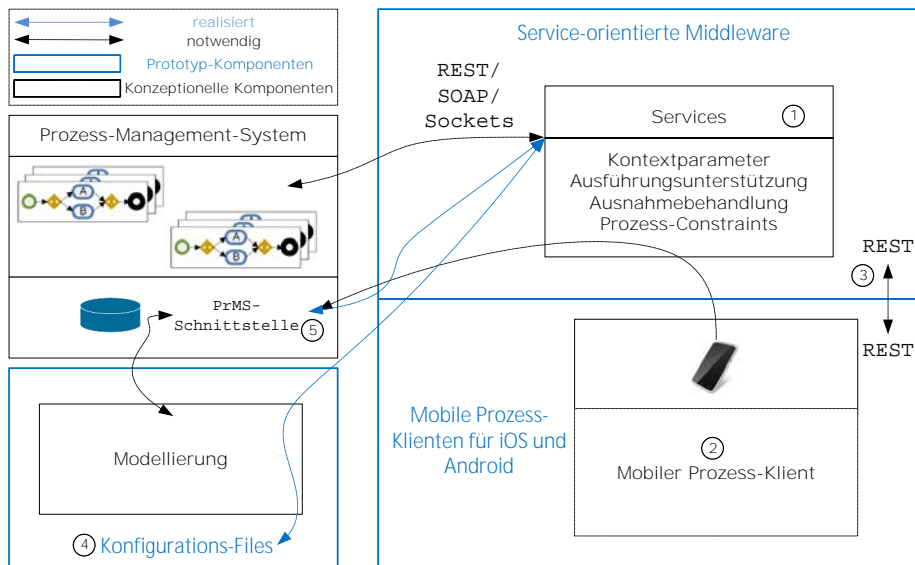


Abbildung 12.1: Konzeption des Prototyps

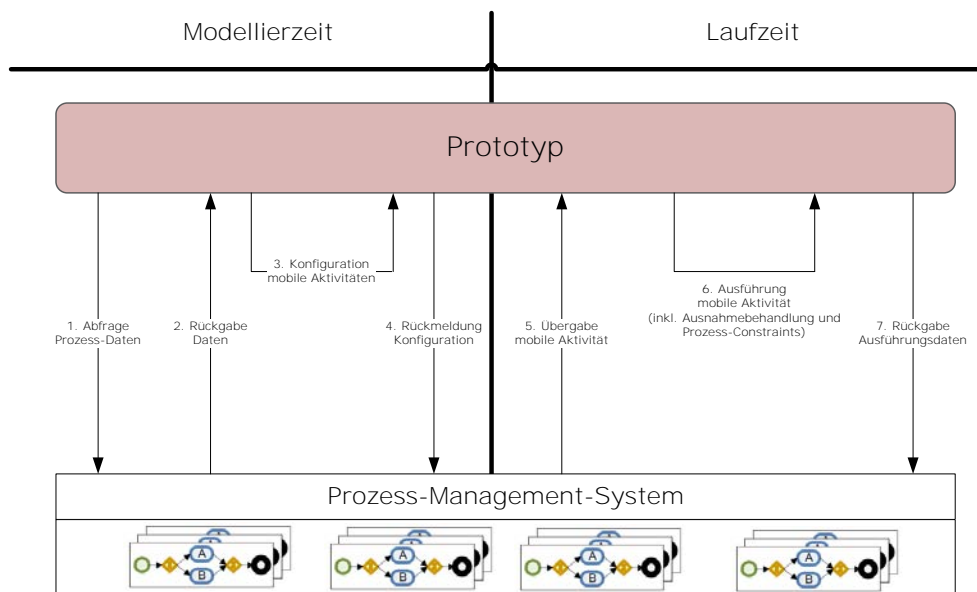


Abbildung 12.2: Integrationsablauf zwischen Prototyp und PrMS

12.2 Service-orientierte Middleware

Die eingangs erwähnte Service-orientierte Middleware wurde mithilfe des Play Frameworks [Pla14] entwickelt (vgl. Abb. 12.3). Dieses zeichnet sich durch einen leistungsfähigen Applikationsserver aus. Anwendungen können wahlweise in Java oder Scala [Sca14] entwickelt werden.

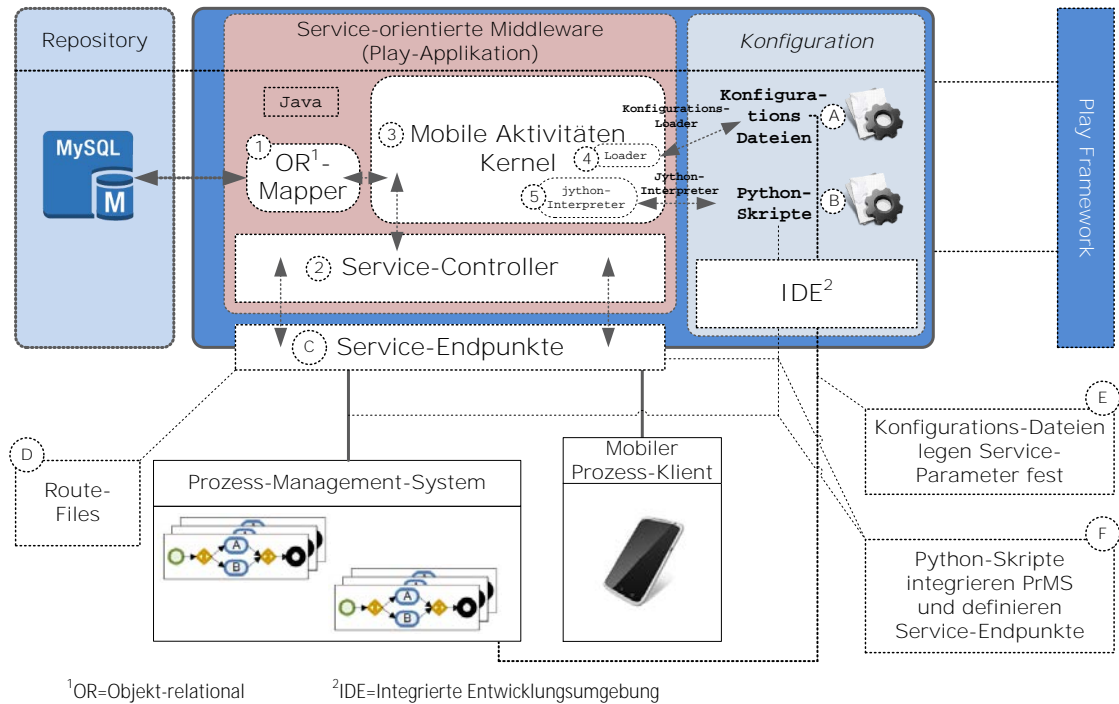


Abbildung 12.3: Service-orientierte Middleware

Die Service-orientierte Middleware umfasst fünf Komponenten (vgl. Abb. 12.3 ①-⑤):

1. Ein OR-Mapper [orm14] bewerkstelligt zwischen der Play-Applikation und dem Repository die Datenkonvertierung zwischen Java-Objekten und Datenbank-Relationen.
2. Eine Sammlung von *Service-Controllern*, welche die Logik für Service-Endpunkte implementiert, die wiederum in sog. Route-Files festgelegt werden. Die Service-Endpunkte regeln die Kommunikation mit den Prozess-Klienten auf den Smart-Mobilgeräten, ebenso wie mit dem *PrMS*.
3. Der *Kernel* des Prototyps realisiert die in dieser Arbeit vorgestellten Konzepte, zum Beispiel die Ausnahmebehandlung für mobile Aktivitäten.
4. Der *Konfigurations-Loader* verwaltet die Konfiguration der Parameter zur Ausführung mobiler Aktivitäten. Hier wird zum Beispiel der Timeout festgelegt, nach dessen Ablauf eine Ausnahmebehandlung für mobile Aktivitäten gestartet werden soll.
5. Ein *Jython-Interpreter* lädt Python-Konfigurationsskripte, welche die Kommunikation mit dem *PrMS* ermöglichen. Python-Skripte haben sich als flexibel erwiesen, um verschiedene Arten der Kommunikation zu realisieren. Dieser Ansatz wurde gewählt, da der Prototyp mit einer Vielzahl von *PrMS* verknüpft werden können soll. Die Python-Skripte realisieren die im Kontext der Service-Controller erwähnten Service-Endpunkte.

12.3 Schnittstelle zum Prozess-Management-System

Dieser Abschnitt behandelt die Schnittstelle des Prototyps zum *PrMS* (vgl. die Markierungen © und ④ in Abb. 12.3). Die Kommunikation zwischen Schnittstelle (den Service-Endpunkten) und *PrMS* wird über Python-Skripte realisiert und mittels sog. Route-Files konfiguriert (vgl. Abb. 12.4①). Mit den Python-Skripten werden zwei Aspekte implementiert, die eine flexible Integration existierender *PrMS* ermöglichen: (1) die Konvertierung des im Kernel verwendeten Datenformats ② in das Datenformat des *PrMS* ③ (und umgekehrt); (2) die Konfiguration des Kommunikationspatterns ④ für das jeweilige *PrMS*. Bezogen auf das Kommunikationspattern kann zum Beispiel konfiguriert werden, ob ein Pull- oder Push-Verfahren zwischen Prototyp und *PrMS* zum Einsatz kommen soll.

Zur Laufzeit werden die Python-Skripte vom jython-Interpreter geladen, interpretiert und in Java-Binärcode umgesetzt. Letzterer wird vom Kernel verwendet, um mit dem *PrMS* Daten auszutauschen. Ferner ist im Kernel ein Datenbus realisiert, der die Kommunikation zwischen den einzelnen Kernel-Modulen bewerkstelligt. Beispiele für Module sind die Ausnahmebehandlung oder die kontextbezogene Endanwenderzuteilung bei der Ausführung mobiler Aktivitäten.

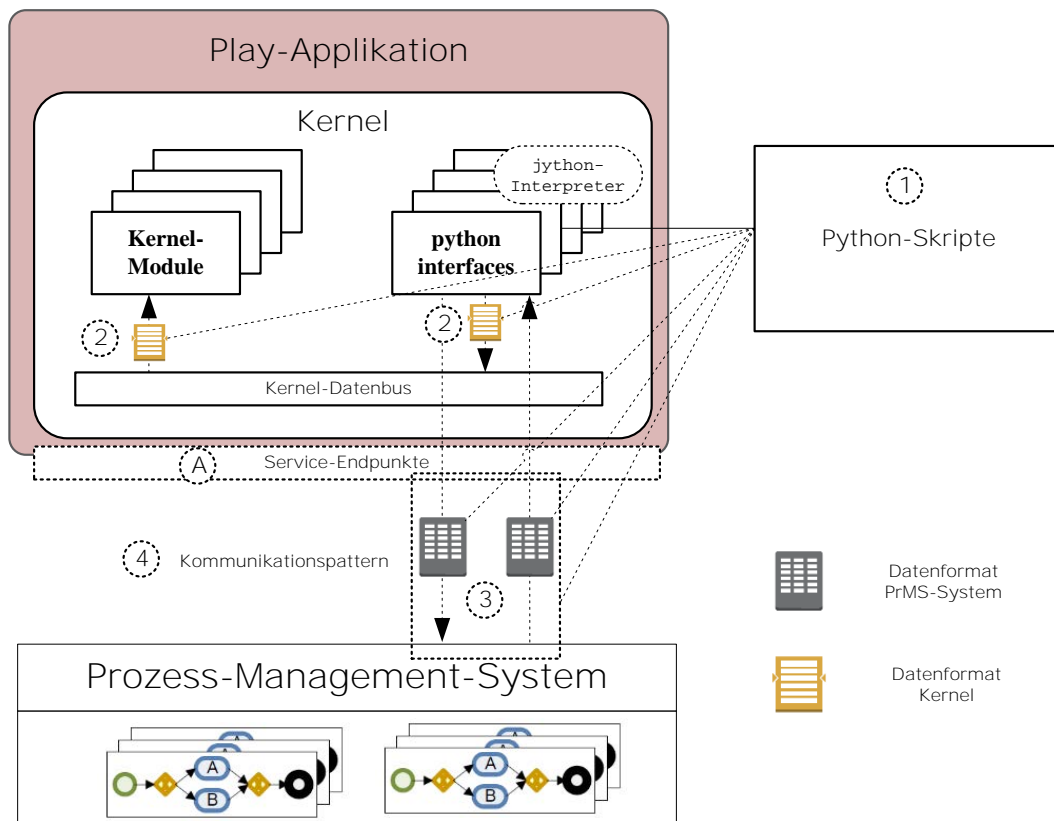


Abbildung 12.4: Verwendungszweck der Python-Skripte

Das Konzept für die Zuordnung von Anfragen bzw. Service-Endpunkten zu Python-Skripten ist dem Konzept von Java nachempfunden, mittels dem sich strukturierte Datentypen aus objektrelationalen Datenbanken in Java-Datentypen dynamisch konvertieren lassen [SQL14a,

SQL14b]. Insbesondere die Realisierung der *java.util.Map* wurde übernommen (vgl. Listing 12.1).

Listing 12.1: *java.util.Map* und *SQLData-Java-Interface*

```

1 // setup mappings for the connection
2 try {
3     java.util.Map map = con.getTypeMap();
4     map.put("S.RESIDENCE", Class.forName("Residence"));
5     map.put("S.FULLNAME", Class.forName("Fullname"));
6     map.put("S.PERSON", Class.forName("Person"));
7 }
8 catch (ClassNotFoundException ex) {}

```

Die Informationen im MAP-Objekt entsprechen den Python-Skripten. Diese werden dynamisch, je nach Anfrage, vom jython-Interpreter geladen und in Java-Binärkode umgesetzt.

12.4 Schnittstelle zum mobilen Prozess-Klienten

Die Kommunikation zwischen *PrMS* und mobilem Prozess-Klienten wird mittels REST realisiert und ebenfalls mittels Python-Skripten implementiert. So existieren Python-Skripte, welche REST-Schnittstellen zur *Authentifizierung* und *Aktivitätenausführung* bereitstellen. Eine Diskussion aller REST-Schnittstellen würde den Rahmen sprengen. Daher sollen hier exemplarisch ausgewählte REST-Schnittstellen für die *Authentifizierung* vorgestellt werden (vgl. Tabelle 12.1).

Endpunkt	Methode	Beschreibung
<i>/auth/login</i>	<i>POST</i>	Endpunkt zur Anmeldung mobiler Endanwender.
<i>/auth/logou</i>	<i>GET</i>	Endpunkt zur Abmeldung mobiler Endanwender.
<i>/auth/dam</i>	<i>POST</i>	Schnittstelle zur Verwaltung von <i>Device Alive Messages</i> . Diese Schnittstelle verwendet der mobile Prozess-Klient, um seinen Online-Status mitzuteilen.

Tabelle 12.1: REST-Schnittstelle zur Autorisierung von mobilen Endanwendern

Zuletzt werden die Route-Files erläutert, die ebenfalls zur Schnittstelle gehören (vgl. Abb. 12.3①). Sie regeln die Konfiguration zwischen Service-Endpunkten und -Controllern. Listing 12.2 zeigt das Route-File für die REST-Methoden aus Tabelle 12.1.

Listing 12.2: Route-File für die Autorisierungs REST-Methoden

```

1 GET /inittest/:testname controllers.TestInitService.initTest(testname:String)
2
3
4 ##Authorisation
5 POST /config/authrequest controllers.config.AuthenticationService.authReq()
6 POST /config/authfinal controllers.config.AuthenticationService.authFinal()
7 GET /config/logout controllers.config.AuthenticationService.logout(token:String)

```

12.5 Kernel

Dieser Abschnitt stellt drei ausgewählte Funktionen des Kernels vor.

12.5.1 Zuteilung mobiler Aktivitäten

Die Zuteilung mobiler Aktivitäten ist wie folgt realisiert (vgl. Abb. 12.5): Der *Activation Service* des Kernels erfragt beim *PrMS* anstehende mobile Aktivitäten. Diese werden durch den *Aktivitäten-Manager* an Endanwender zugeteilt. Dazu spaltet der *Aktivitäten-Manager* für jede auszuführende mobile Aktivität einen *Thread* ab, der die Zuteilung und Ausführung bewerkstelligt. Ferner werden von diesem Thread notwendige Prozessdaten geladen. Zuletzt werden mithilfe der vorgestellten Algorithmen geeignete mobile Endanwender bestimmt und deren Worklists mithilfe des *Worklist-Managers* befüllt. Die Bestimmung geeigneter mobiler Endanwender erfolgt mittels Filtern, welche die in den Kapiteln 8, 9 und 10 vorgestellten Algorithmen implementieren (vgl. Abb. 12.5①). Sobald ein mobiler Endanwender sich für eine mobile Aktivität entschieden hat, kann diese gestartet und ausgeführt werden.

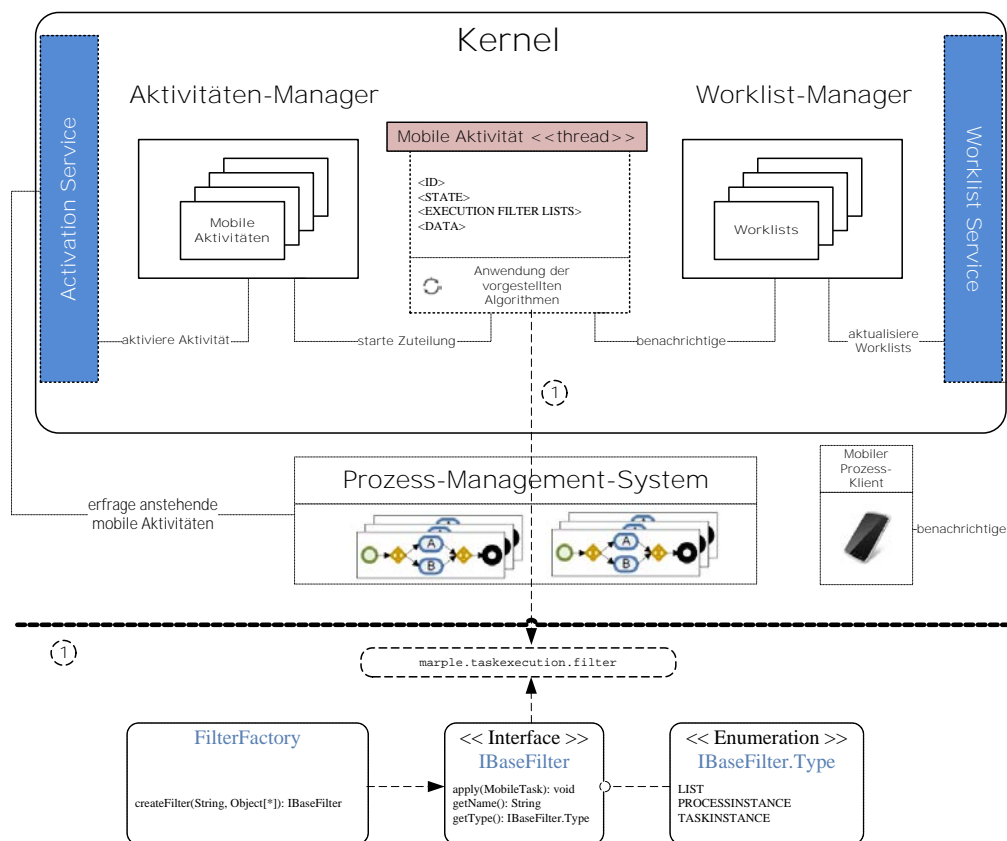


Abbildung 12.5: Zuteilung mobiler Aktivitäten

12.5.2 Ausführung mobiler Aktivitäten

Die Ausführung mobiler Aktivitäten beinhaltet folgende Schritte (vgl. Abb. 12.6): Sobald eine mobile Aktivität durch einen mobilen Endanwender gestartet wurde, muss das betreffende Smart-Mobilgerät seinen Online-Status regelmäßig mit *DAM-Messages*¹ an den Kernel melden.

¹DAM steht für Device Alive Message.

Solange das Smart-Mobilgerät offline bleibt, wird die mobile Aktivität ausgeführt, bis sie ordnungsgemäß beendet wurde. Ergibt sich jedoch eine Änderung des Online-Status, registriert der *Geräte-Manager* dies. Er benachrichtigt dann den *Aktivitäten-Manager*², der Statusänderungen verarbeitet und dem ausführenden Thread einer mobilen Aktivität diese Änderung mitteilt. Bei Statusänderungen werden Worklists betroffener mobiler Endanwender durch den *Worklist-Manager* aktualisiert (z.B. bei einer Delegation).

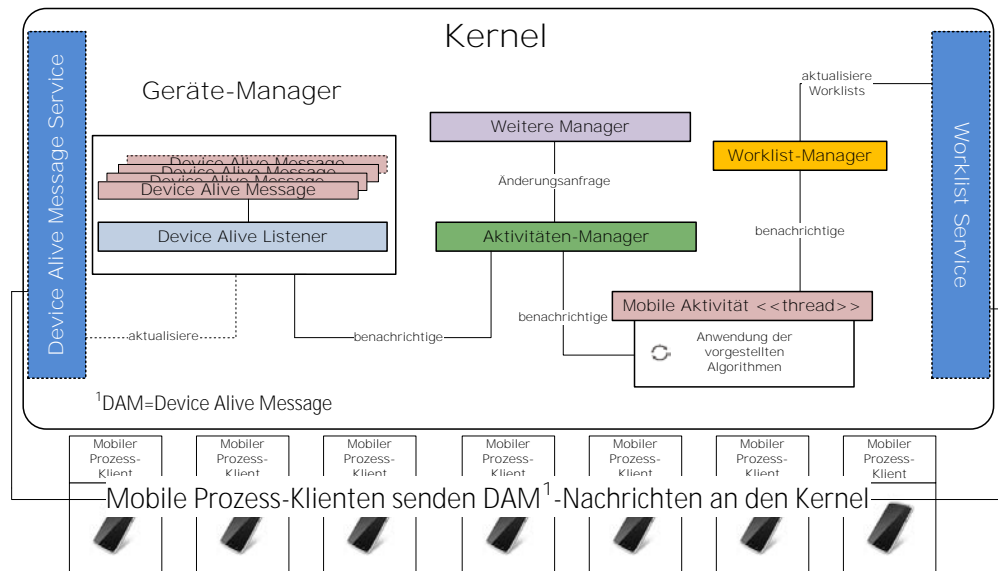


Abbildung 12.6: Ausführung mobiler Aktivitäten

12.5.3 Beenden mobiler Aktivitäten

Es existieren zwei Varianten, die zum Beenden einer mobilen Aktivität führen können. Bei der ersten meldet der für die mobile Aktivität verantwortliche Thread, dass die Aktivitätenausführung beendet ist. D.h. entweder wurde die Aktivität ordnungsgemäß beendet oder sie wurde abgebrochen. Bei der zweiten Variante meldet ein Manager (z.B. der für die Prozessinstanz verantwortliche), dass die Aktivität beendet werden soll, etwa wenn eine Prozessinstanz durch das *PrMS* komplett abgebrochen wurde. Ist die Aktivität beendet, wird dies an den *Aktivitäten-Manager* gemeldet. Außerdem wird der *Worklist-Manager* benachrichtigt, der daraufhin die Worklists betroffener mobiler Endanwender aktualisiert. Zuletzt wird das Beenden einer mobilen Aktivität an das *PrMS* zu Log-Zwecken rückgemeldet (vgl. Abb. 12.7①).

12.6 Repository

Abbildung 12.4 zeigt das realisierte Repository auf Basis einer MySQL-Datenbank (vgl. Abb. 12.8). Ferner wurden Skripte, Trigger und Stored Procedures auf der MySQL-Datenbank implementiert, um das Repository zu realisieren. Letzteres speichert Daten zum Smart-Mobilgerät, zu

²Auch weitere Manager können den *Aktivitäten-Manager* während der Ausführung kontaktieren, um eine Status-Änderung der mobilen Aktivität herbeizuführen.

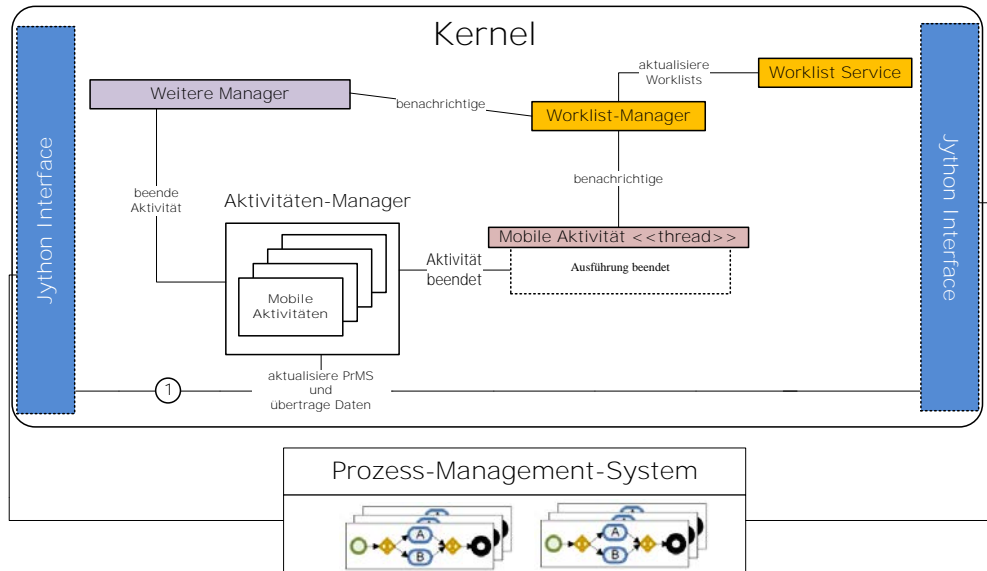


Abbildung 12.7: Beenden einer mobilen Aktivität

den mobilen Aktivitäten, zu den Eigenschaften mobiler Endanwender sowie zu den Prozessen, in deren Kontext die mobilen Aktivitäten ausgeführt werden (s. Abb. 12.8 für die verwalteten Daten).

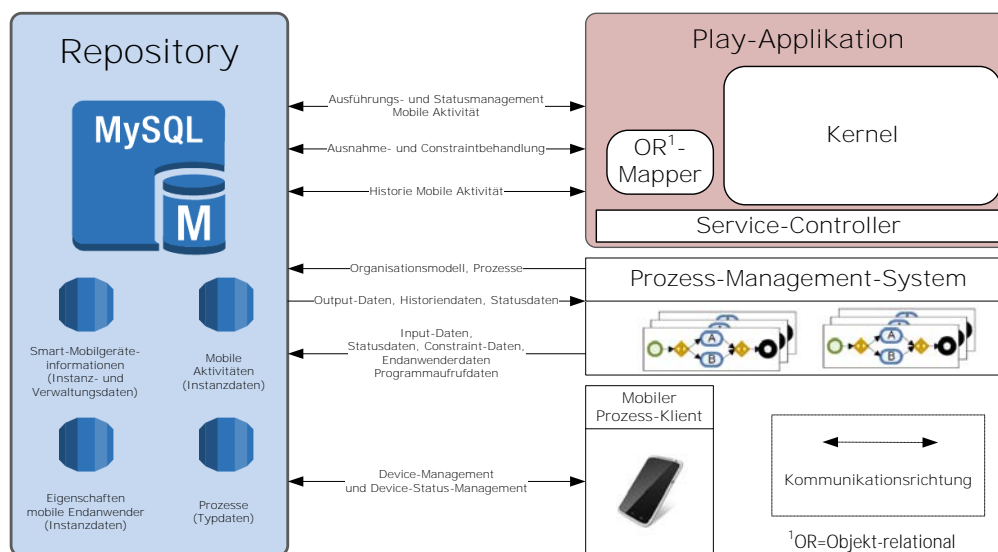


Abbildung 12.8: Repository mit gespeicherten Daten

12.7 Mobiler Prozess-Klient

Bevor der realisierte Prozess-Klient vorgestellt wird, sollen einige Vorüberlegungen angestellt werden.

Erstens muss diskutiert werden, welche Art der mobilen Anwendungsentwicklung gewählt werden soll (vgl. Kapitel 3). Der mobile Prozess-Klient wurde als native Anwendung realisiert. Dies hatte zwei Gründe. Erstens lässt sich dadurch die freundlichste Benutzerschnittstelle entwickeln. Zweitens lässt sich auf diese Art und Weise das Ausführungsprotokoll zwischen Kernel und Prozess-Klient effizient und robust realisieren.

Zweitens muss entschieden werden, welches mobile Betriebssystem unterstützt werden soll. Konkret wurden Prototypen für Android [And14] und iOS [iOS14] entwickelt. Die Implementierung für das Ausführungsprotokoll wurde primär in Android realisiert, da es gut mit dem Kernel harmoniert, da beiderseits Java zum Einsatz kommt. Nichtsdestotrotz wurde auch eine Beispiel-Implementierung unter iOS realisiert.

Drittens sollen Ansätze identifiziert werden, die als gute Grundlage für das Benutzerschnittstelle auf dem mobilen Prozess-Klienten dienen. In [MOL⁺08, Ara14, Dö14, Lan12, PLRH12, PMLR14] wird weiters Windows Phone [win14] untersucht, sodass alle wichtigen mobilen Betriebssysteme abgedeckt sind.

Viertens mussten mobile Betriebssysteme daraufhin untersucht werden, ob auf allen die Hardware-Parameter des in dieser Arbeit vorgestellten mobilen Kontexts erfasst werden können. Für Android, iOS und Windows Phone konnte dies nachgewiesen werden.

Schließlich wurde untersucht, ob relevante Aspekte im Kontext des Ausführungsprotokolls (z.B. Verdrängen mobiler Apps in den Hintergrund) auf allen mobilen Betriebssystemen gewährleistet werden können. Auch dies konnte gezeigt werden.

Die Abbildungen 12.9 und 12.10 zeigen ausgewählte Benutzerschnittstellen der Prozess-Klienten auf Android und iOS. Abbildung 12.9 zeigt den Anmeldevorgang und die Worklist anstehender mobiler Aktivitäten.

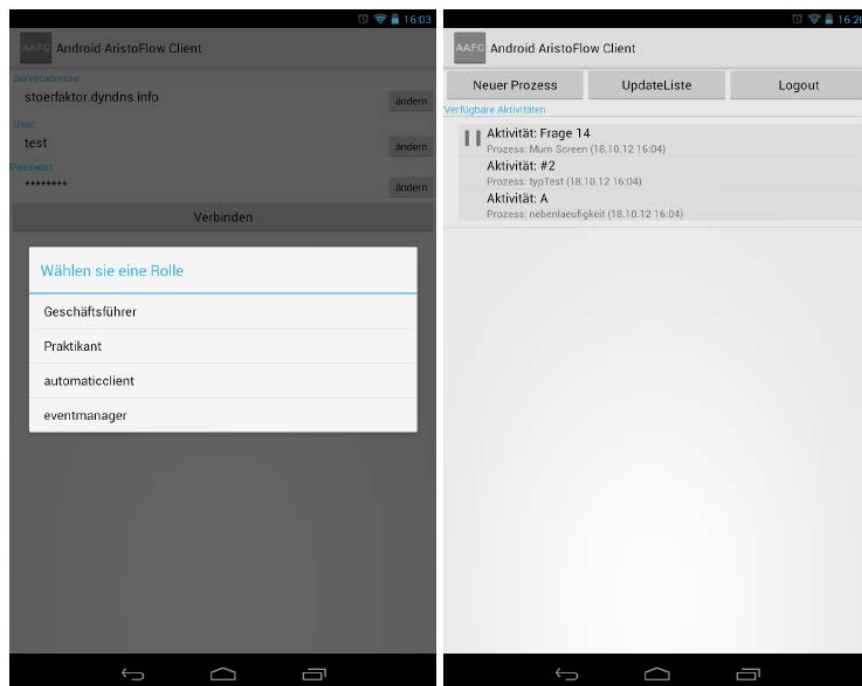


Abbildung 12.9: Mobiler Prozess-Klient auf Android

Abbildung 12.10 wiederum illustriert wie die iOS-Benutzerschnittstelle des Prozess-Klienten entworfen wurde. Es wurde mit Handskizzen begonnen, die mit Endanwendern diskutiert wurden. Das hieraus erstellte Feedback floss dann in das endgültige Design der Benutzerschnittstelle mit ein.

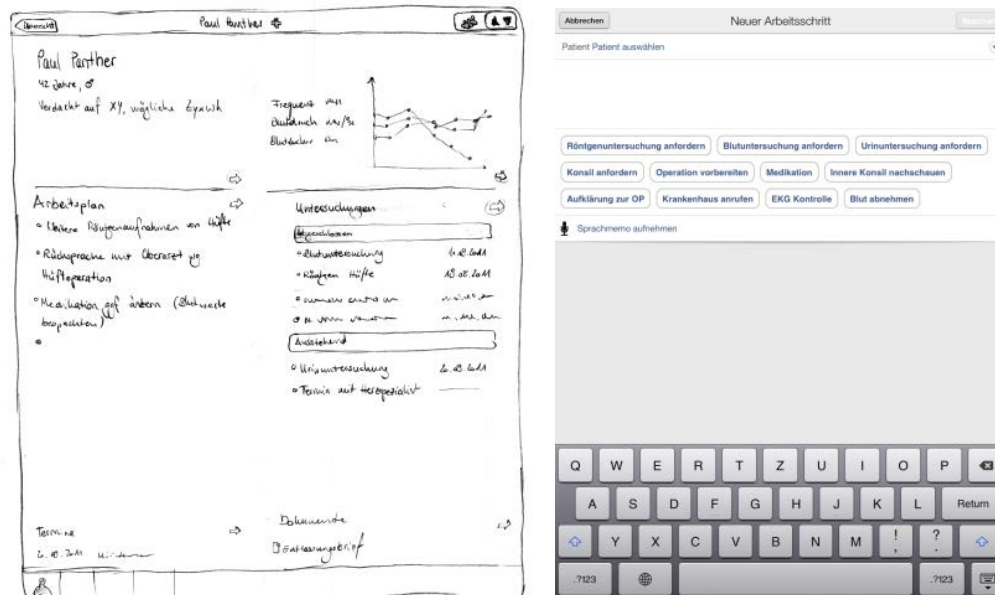


Abbildung 12.10: Mobiler Prozess-Klient auf iOS

12.8 Weitere Aspekte

Abschließend sollen vier Aspekte des Prototyps diskutiert werden:

- Da der Prototyp mit einem *PrMS* verbunden wird, muss sichergestellt werden, dass mobile Aktivitäten tatsächlich vom Prototypen und damit mobil ausgeführt werden. Dies wird erreicht, indem der Prototyp alle Aktivitäten ausführt. Normalerweise würden nicht-mobile Aktivitäten direkt vom *PrMS* ausgeführt.
- Es muss mit Ausführungsdaten mobiler Aktivitäten, die existierende *PrMS* nicht kennen, etwa bei einer durchgeführten Delegation, umgegangen werden. Diese Daten werden nicht zurück an das *PrMS* übermittelt, sondern nur im Repository des Prototyps gespeichert.
- Die Integration des Prototypen mit verschiedenen *PrMS* muss möglich sein. Folgende *PrMS* wurden betrachtet: (1) *Activiti* [Rad12, Cum07], das unter der Open Source Lizenz steht; (2) *AristaFlow* [LKRD10, RDRM⁺09b, LRD10a, LKRD10, LRD10b], das sich durch innovative Funktionen auszeichnet, insbesondere hinsichtlich robuster Ausführung und Prozessflexibilität. Diese beiden *PrMS* ließen sich mit dem Prototypen problemlos integrieren.
- Zuletzt geht es um die Modellierung mobiler Aktivitäten. So wurde bereits im Kapitel 4 diskutiert, dass keine dedizierte Modellierungskomponente für mobile Aktivitäten entwi-

ckelt werden soll. Die Modellierung im Prototyp erfolgt durch Konfigurationsdateien, die für den Einsatzzweck des Prototyps ausreichend sind. Aktuell wird für zukünftige Projekte eine Modellierungskomponente entwickelt [PRBA15], welche mithilfe von 3D-Modellen die Konfiguration mobiler Aktivitäten direkt auf dem Smart-Mobilgerät ermöglicht (s. Abb. 12.11).

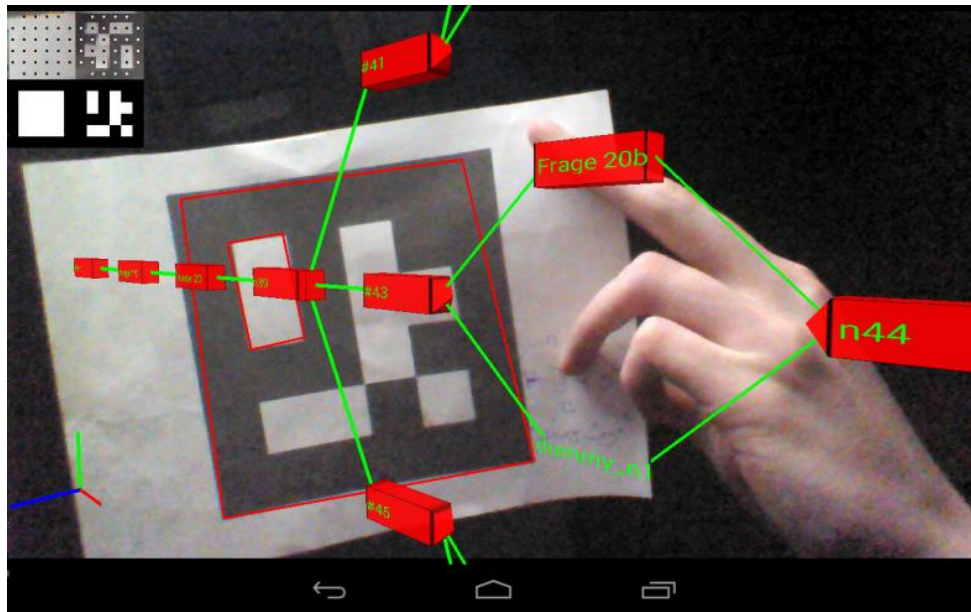


Abbildung 12.11: Konfiguration mobiler Aktivitäten mithilfe von 3D-Modellen [PRBA15]

12.9 Zusammenfassung

Dieses Kapitel hat die prototypische Implementierung der entwickelten Konzepte vorgestellt. Insbesondere wurde gezeigt, wie die Implementierung des Prototyps erfolgte und wie sich die jeweiligen Entwicklungsentscheidungen begründeten. Ausgewählte Komponenten des Prototyps wurden vorgestellt sowie deren Limitationen diskutiert.

Nicht weil es so schwer ist wagen wir es nicht, sondern weil wir es nicht wagen, ist es so schwer.

Lucius Annaeus Seneca, (1 - 65 n. Chr.)

13

Praktische Anwendung

Der entwickelte Prototyp hat die technische Machbarkeit der Konzepte für mobile Aktivitäten gezeigt. Weiters muss die praktische Anwendbarkeit des Prototyps demonstriert werden. Nur so kann der Nachweis erbracht werden, dass die entwickelten Konzepte auch praktikabel sind. Dieses Kapitel zeigt die praktische Anwendung des Prototyps in Fallstudien unterschiedlicher Anwendungsdomänen. Abschnitt 13.1 stellt die Fallstudien vor. In Abschnitt 13.2 wird eine Fallstudie zur Lagerverwaltung im Detail behandelt. Abschnitt 13.3 wiederum diskutiert eine Fallstudie im Rettungsdienst. Schließlich behandelt Abschnitt 13.4 eine Klinische Studie in der Psychologie. Abschnitt 13.5 diskutiert sonstige Aspekte, während Abschnitt 13.6 das Kapitel zusammenfasst.

13.1 Einleitung

Der Prototyp wurde in drei Fallstudien angewandt. Diese wiederum entstammen den Anwendungsdomänen Logistik, Gesundheitswesen und Klinische Studien (vgl. Tabelle 13.1).

Es sei an dieser Stelle auf drei Aspekte bei der Durchführung der Fallstudien hingewiesen. Erstens wurden in allen Fallstudien Prozessmodelle erhoben. Dies wurde analog zu den Fallstudien aus Kapitel 2 bewerkstelligt. Zweitens wurde AristaFlow [LKRD10] in allen Fallstudien als *PrMS* verwendet. Da in keiner der Fallstudien ein *PrMS* im Einsatz war, mussten keine bestehenden *PrMS* berücksichtigt werden. Drittens wurden in allen Fallstudien die Smart-Mobilgeräte der Mitarbeiter genutzt. Dies waren sowohl iOS- als auch Android-Geräte. Da für Windows Phone kein mobiler Prozess-Klient entwickelt wurde, konnten diese Smart-Mobilgeräte nicht eingebunden werden. Allerdings stellt dies keine Einschränkung dar, da lediglich in einer Fallstudie ein Windows Phone vorlag.

13.2 Fallstudie 1: Lagerverwaltung

In einem Großlager eines Logistikunternehmens wurde untersucht, welche Tätigkeiten konkret von Lagerarbeitern durchgeführt werden. Nachdem Prozessmodelle zu den identifizierten Tä-

Fallstudie zu	Beschreibung	Merkmale
Lagerverwaltung	Tätigkeiten eines Großlagers wurden daraufhin analysiert, ob diese als mobile Aktivität ausgeführt werden können.	Hohe Dynamik durch viele Abweichungen, viele Endanwender, umfangreiche Sensormessungen, umfangreiche Kontextinformationen
Rettungsdienst	Tätigkeiten beim Rettungseinsatz und bei der Übergabe des Patienten in der Klinik wurden daraufhin analysiert, ob diese als mobile Aktivität ausgeführt werden können.	Hohe Dynamik durch viele Abweichungen, umfangreiche Sensormessungen, umfangreiche Kontextinformationen, potentiell viele Ursachen für Ausnahmen (z.B. Netzausfälle), Offline-Situationen, umfangreiche Dateneingabe
Klinische Studien	Tätigkeiten bei klinischen Studien wurden daraufhin untersucht, ob diese als mobile Aktivität ausgeführt werden können.	Offline-Situationen, viele Endanwender, viele Sensormessungen

Tabelle 13.1: Einsatz des Prototyps in Fallstudien

tigkeiten und deren Abfolge entwickelt werden konnten, wurden Prozessaktivitäten daraufhin untersucht, ob sie sich für eine mobile Ausführung eignen (vgl. Tabelle 13.2). Weiters konnte festgestellt werden, dass sich die Aktivitäten vier Ausführungsbereichen zuordnen lassen (vgl. Tabelle 13.2). Das Ergebnis der Anwendung der vorgestellten Konzepte mobiler Aktivitäten ist den Tabellenspalten 3-7 zu entnehmen. Vor allem in hektischen Betriebszeiten waren die kontextbezogene Endanwenderzuteilung und die Ausnahmebehandlung aus Sicht der Endanwender eine Arbeitserleichterung. Hierbei erwiesen sich der Ausführungsort und der Formfaktor bei der Zuteilung mobiler Aktivitäten als wichtige Parameter. Die Offline-Ausführung mobiler Aktivitäten wurde vor allem dann nötig, wenn sich Endanwender außerhalb der Firma (z.B. Außendienst) befanden. Das Konzept der Offline-Aktivität überzeugte die Endanwender ebenfalls. Die Ausnahmebehandlung wurde generell begrüßt, hauptsächlich kam aber nur die Delegation zum Einsatz, wohingegen das Backup nur selten vonnöten war. Es kann festgehalten werden, dass die Konzepte mobiler Aktivitäten als Arbeitserleichterung empfunden wurden.

13.3 Fallstudie 2: Rettungsdienst

Die zweite Fallstudie untersuchte für Einsätze im Rettungsdienst (Krankentransporte, Rettungseinsätze), die jeweils vom Rettungsdienst durchgeführten Tätigkeiten. Szenarien im Rettungsdienst sind aus IT-Sicht herausfordernd. So müssen gesetzliche Auflagen beachtet werden, Einsätze finden an Orten statt, in denen nur das Telefonnetz vorausgesetzt werden kann, und Entscheidungen müssen in den meisten Fällen schnell gefällt werden. Dennoch konnten die konkreten Tätigkeiten im Zusammenspiel mit dem Personal des Rettungsdiensts identifiziert werden. Ferner existieren für viele Abläufe bereits papierbasierte Prozesse (z.B. Prozess für Megacode [meg14]), was die Identifikation von Tätigkeiten erleichterte.

Aktivität	Beschreibung	KE	AB	PC	FF	OA
Innerhalb des Lagers						
Artikelbestand	Warenbestand überprüfen	✓	✓	✓	<i>B</i>	<i>X</i>
Regalbestand	Regalbestand überprüfen	✓	✓	✓	<i>B</i>	<i>X</i>
Einlagern	Angelieferter Artikel einlagern	✓	✓	✓	<i>S</i>	<i>X</i>
Entnehmen	Eingelagerte Artikeln entgegennehmen	✓	✓	✓	<i>S</i>	<i>X</i>
Informationsabruf	Artikelinformationen abrufen	–	–	–	<i>B</i>	<i>X</i>
Informationsabruf	Regalinformationen abrufen	–	–	–	<i>B</i>	<i>X</i>
Reklamation	Reklamation entgegennehmen	✓	✓	✓	<i>B</i>	<i>X</i>
Reklamation	Reklamation durchführen	✓	✓	✓	<i>B</i>	<i>X</i>
Außerhalb des Lagers						
Überprüfung	Angelieferte Waren überprüfen	✓	✓	✓	<i>S</i>	✓
Entgegennahme	Angelieferte Waren entgegennehmen	✓	✓	✓	<i>S</i>	✓
Aushändigung	Waren an LKW abgeben	✓	✓	<i>X</i>	<i>S</i>	✓
Aushändigung	Waren an Abholer abgeben	✓	✓	<i>X</i>	<i>S</i>	✓
Aushändigung	Waren durch LKW vor Ort abgeben	✓	<i>X</i>	✓	<i>B</i>	✓
Reklamation	Reklamation entgegennehmen	✓	✓	✓	<i>B</i>	✓
Reklamation	Reklamation durchführen	✓	✓	✓	<i>B</i>	<i>X</i>
Schnittstelle Verwaltung						
Lieferschein	Lieferschein zur Warenabgabe erstellen	✓	✓	✓	<i>B</i>	<i>X</i>
Lieferschein	Lieferschein zur Warenannahme erstellen	✓	✓	✓	<i>B</i>	<i>X</i>
Reklamation	Reklamation entgegennehmen	✓	✓	✓	<i>B</i>	<i>X</i>
Reklamation	Reklamation durchführen	✓	✓	✓	<i>B</i>	<i>X</i>
Kunden- und Großhandelsgespräche						
Großhändler	Artikelinformationen abrufen	–	–	✓	<i>B</i>	✓
Großhändler	Bestelldaten abrufen	–	–	✓	<i>B</i>	✓
Kundengespräche	Artikelinformationen abrufen	✓	✓	–	<i>B</i>	✓
Kundengespräche	Bestellung vor Ort annehmen	✓	✓	✓	<i>B</i>	✓
Reklamation	Reklamation entgegennehmen	✓	✓	✓	<i>B</i>	✓
Reklamation	Reklamation durchführen	✓	✓	✓	<i>B</i>	<i>X</i>

KE=Kontextbezogene Endanwenderzuteilung, AB=Ausnahmebehandlung, PC=Prozess-Constraints
FF=Formfaktor, OA=Offline-Aktivität, ✓=trifft zu, X=trifft nicht zu, –=irrelevant
S=Smartphone, T=Tablet, B=Beides

Tabelle 13.2: Mobile Aktivitäten zur Fallstudie in der Lagerverwaltung

Analog zur Fallstudie 1 wurden zusätzlich Prozessmodelle und mobil auszuführende Tätigkeiten identifiziert (vgl. Tabelle 13.3). Ebenso konnten verschiedene Ausführungsbereiche für die mobilen Aktivitäten identifiziert werden (vgl. Tabelle 13.3). Die Spalten 3-7 der Tabelle 13.3 geben wieder das Ergebnis der Anwendung der Konzepte mobiler Aktivitäten an.

Die Anwendung der *BoD*- und des *Card*-Constraints waren für Fallstudie 2 relevant. Gesetzlich entsteht oftmals die Notwendigkeit im Rettungsdienst, zwei oder mehr Tätigkeiten vom selben Endanwender durchführen zu lassen. Daher wurde der *BoD*-Constraint oftmals benötigt, insbesondere auch im Kontext mobiler Tätigkeiten. Im Zusammenhang mit gesetzlichen Dokumentationsanforderungen ist wiederum der *Card*-Constraint von Bedeutung, da Dokumentationsaufgaben oft mehrfach durchgeführt werden müssen.

Aktivität	Beschreibung	KE	AB	PC	FF	OA
Rettungsannahme und Verlassen des Standorts						
Einsatzverarbeitung	Einsatz entgegennehmen	✓	–	–	<i>S</i>	–
Einsatzverarbeitung	Einsatz ablehnen	✓	–	–	<i>S</i>	–
Einsatzverarbeitung	Einsatz übernehmen	✓	–	–	<i>S</i>	–
Spezialequipment	Spezialequipment mitnehmen	–	–	✓	<i>S</i>	✓
Navigation	Navigation starten	–	–	✓	<i>S</i>	✓
Fahrt zum Patienten, Patientenbehandlung, Fahrt zur Patientenübergabe						
Unfallort	Unfallort suchen	–	–	✓	<i>S</i>	✓
Leitstelle	Leitstelle kontaktieren	–	–	✓	<i>S</i>	✓
Ankunft	Unfallort erreichen	–	–	✓	<i>S</i>	✓
Rettungsmittel	Rettungsmittel anfordern	–	–	✓	<i>S</i>	✓
Spezialkräfte	Spezialkräfte (Polizei, THW) anfordern	–	–	✓	<i>S</i>	✓
Abbruch	Einsätze abbrechen	–	–	✓	<i>S</i>	✓
Änderung	Einsätze ändern	–	–	✓	<i>S</i>	✓
Abfrage	Medizinische Informationen abfragen	–	–	✓	<i>S</i>	✓
Patientenaufnahme	Patientendaten ermitteln	✓	–	✓	<i>S</i>	✓
Patiententherapie	Therapiemaßnahmen dokumentieren	✓	✓	✓	<i>S</i>	✓
Kontaktaufnahme	Weiterbehandelnde Einrichtung, Leitstelle oder andere Rettungsmittel kontaktieren	–	–	✓	<i>S</i>	✓
Patientenübergabe	Patienten an andere Rettungsmittel übergeben	–	–	✓	<i>S</i>	✓
Übergabe des Patienten, Rückfahrt und Abschluss des Rettungseinsatzes						
Ankunft	Weiterbehandelnder Einrichtung erreichen	–	–	✓	<i>S</i>	✓
Übergabe	Patienten an weiterbehandelnde Einrichtung übergeben	–	–	✓	<i>S</i>	✓
Dokumentation	Patientenübergabe dokumentieren	–	–	✓	<i>S</i>	✓
Rückfahrt	Zum Standort zurückfahren	–	–	✓	<i>S</i>	✓
Verarbeitung	Neuen Einsatz annehmen	–	–	✓	<i>S</i>	✓
Verarbeitung	Neuen Einsatz ablehnen	–	–	✓	<i>S</i>	✓
Standort	Standort erreichen	–	–	✓	<i>S</i>	✓
Dokumentation	Am Standort finale Dokumentation anfertigen	✓	✓	✓	<i>B</i>	✓
Auffüllen	Verbrauchte Materialien auffüllen	✓	✓	✓	<i>B</i>	✓
Auffüllen	Dokumentation anfertigen	✓	✓	✓	<i>B</i>	✓
Spezialmaßnahmen	Spezialmaßnahmen (Putzen bei besonderen Krankheiten) durchführen	✓	✓	✓	<i>B</i>	✓
Spezialmaßnahmen	Spezialmaßnahmen (Putzen bei besonderen Krankheiten) dokumentieren	✓	✓	✓	<i>B</i>	✓

KE=Kontextbezogene Endanwenderzuteilung, AB=Ausnahmebehandlung, PC=Prozess-Constraints
FF=Formfaktor, OA=Offline-Aktivität, ✓=trifft zu, X=trifft nicht zu, –=irrelevant
S=Smartphone, T=Tablet, B=Beides

Tabelle 13.3: Mobile Aktivitäten zur Fallstudie im Rettungsdienst

13.4 Fallstudie 3: Mobile Datenerfassung in Studien

Bei dieser Fallstudie wurden in mehreren Klinischen Studien des Fachbereichs Psychologie mobile Tätigkeiten von Psychologen identifiziert und untersucht. Der Kontext solcher Studien kann unterschiedlich sein. Manche von ihnen untersuchen zum Beispiel ein bestimmtes Leiden wie Psychische Störungen nach sexuellem Missbrauch [IRLP⁺13, Sch13]. Andere Studien wiederum untersuchen traumatisierte Kindsoldaten in Entwicklungsländern. Die Heterogenität der Studien, die wechselnden Studienorte, die vielen beteiligten Akteure einer Studie sowie das Arbeiten außerhalb kontrollierter Umgebungen machen Studien sehr interessant für den Einsatz mobiler

Aktivitäten.

Auch in dieser Fallstudie wurden Tätigkeiten analysiert und Prozessmodelle erhoben¹ (vgl. Tabelle 13.4). Abbildung 13.1 zeigt den typischen Ablauf eines Fragebogens der Klinischen Psychologie mit seinen einzelnen Tätigkeiten ①. Unter ② und ③ sind ferner Darstellungen der Benutzerschnittstelle des entsprechenden mobilen Prozess-Klienten zu sehen. Analog zu den Fallstudien 1 und 2 konnten verschiedene Ausführungsbereiche für die mobilen Aktivitäten identifiziert werden (vgl. Tabelle 13.4). Die Spalten 3-7 der Tabelle 13.4 fassen das Ergebnis der Anwendung der Konzepte mobiler Aktivitäten zusammen.

Abschließend sei ein Aspekt zu dieser Fallstudie hervorgehoben. Für alle mobilen Tätigkeiten wurde das gesamte Spektrum der vorgestellten Konzepte für mobile Aktivitäten benötigt. Die Psychologen empfanden deren Anwendung als arbeitserleichternd. Ferner wurde berichtet, dass auf Grundlage mobiler Aktivitätenunterstützung weniger Aufgaben vergessen werden.

Aktivität	Beschreibung	KE	AB	PC	FF	OA
Metatätigkeiten						
Fragebogen	Fragenmodus wählen	✓	✓	✓	<i>B</i>	✓
Fragebogen	Fragebogen wählen	✓	✓	✓	<i>B</i>	✓
Fragebogen	Sprache wählen	✓	✓	✓	<i>B</i>	✓
Fragebogen	Metainformationen eingeben	✓	✓	✓	<i>B</i>	✓
Fragebogen	Optionen auswählen	✓	✓	✓	<i>B</i>	✓
Fragebogen	Fragebogen exportieren	✓	✓	✓	<i>B</i>	✓
Fragentätigkeiten						
Frage	Antwort aufschreiben	✓	✓	✓	<i>B</i>	✓
Frage	Zusatzinformationen aufschreiben	✓	✓	✓	<i>B</i>	✓
Frage	Kamerabild schießen	✓	✓	✓	<i>B</i>	✓
Frage	Tonaufnahme starten	✓	✓	✓	<i>B</i>	✓
Frage	Fragen zusammenfassen	✓	✓	✓	<i>B</i>	✓
Frage	Blockkreuzen durchführen	✓	✓	✓	<i>B</i>	✓
Frage	Fragensprung durchführen mit Rückkehr	✓	✓	✓	<i>B</i>	✓
Frage	Fragensprung durchführen ohne Rückkehr	✓	✓	✓	<i>B</i>	✓
Frage	Frage abbrechen	✓	✓	✓	<i>B</i>	✓
Frage	Frage überspringen	✓	✓	✓	<i>B</i>	✓
Frage	Antwort korrigieren	✓	✓	✓	<i>B</i>	✓
Frage	Antworten anzeigen	✓	✓	✓	<i>B</i>	✓
Frage	Sprache ändern	✓	✓	✓	<i>B</i>	✓
<i>KE=Kontextbezogene Endanwenderzuteilung, AB=Ausnahmebehandlung, PC=Prozess-Constraints</i> <i>FF=Formfaktor, OA=Offline-Aktivität, ✓=trifft zu, X=trifft nicht zu, –=irrelevant</i> <i>S=Smartphone, T=Tablet, B=Beides</i>						

Tabelle 13.4: Mobile Aktivitäten zur Fallstudie Klinischer Studien

13.5 Diskussion

Die Fallstudien haben gezeigt, dass eine generelle Unterstützung mobiler Aktivitäten verschiedene Arbeitserleichterungen für Endanwender mit sich bringt. Ferner erwiesen sich die Konzepte zur kontextbezogenen Endanwenderzuteilung, Ausnahmebehandlung und Unterstützung von Prozess-Constraints als wichtige Maßnahmen, die von Endanwendern auch akzeptiert werden.

¹Inzwischen wurde dazu auch ein eigenes Forschungsprojekt initiiert [que14, Fab12].

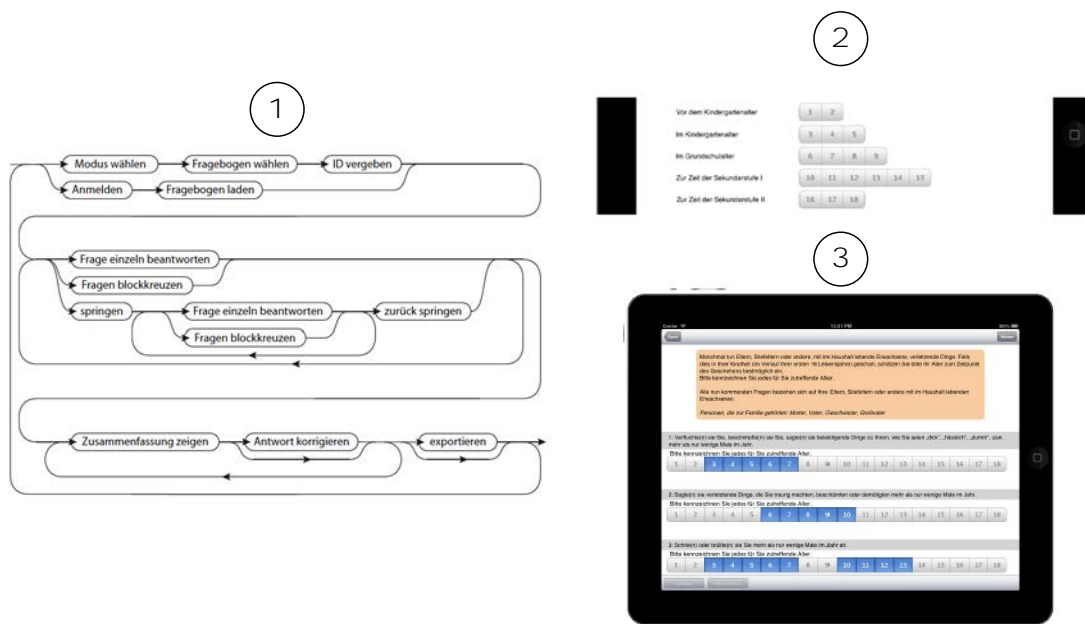


Abbildung 13.1: Fragebogen auf Smart-Mobilgerät

Im Kontext von Ausnahmebehandlungen wurde das Konzept der Delegation als besonders relevant bewertet. Auch das Backup mobiler Aktivitäten war für mobile Szenarien relevant. Hier hat allerdings ein Hinweis auf die Möglichkeit des Systemwechsels (d.h. ein Wechsel der Ausführung der mobilen Tätigkeit auf ein Desktop-System) dazu geführt, dass ein Backup nur in wenigen Ausnahmefällen nötig war.

13.6 Zusammenfassung

Der Prototyp hat sich im praktischen Einsatz in den drei Fallstudien bewährt. Die vorgestellten Konzepte mobiler Aktivitäten konnten dadurch in verschiedenen Anwendungsdomänen unter realen Bedingungen erprobt werden. Weiters wurden sie in allen Fallstudien als arbeitserleichternd und hilfreich durch die Endanwender befunden. Daher konnte durch die Anwendung des Prototyps die praktische Relevanz und Anwendbarkeit der vorgestellten Konzepte für mobile Aktivitäten nachgewiesen werden.

Teil V

Fazit

Der höchste Lohn für unsere Bemühungen ist nicht das, was wir dafür bekommen, sondern das, was wir dadurch werden.

John Ruskin (1819-1900)

14

Zusammenfassung und Ausblick

Diese Arbeit hat sich das Ziel gesetzt, Aktivitäten eines Prozesses mithilfe von Smart-Mobilgeräten robust ausführen zu können. Um dieses Ziel zu erreichen, mussten komplexe Anforderungen berücksichtigt werden. Da in der Literatur diese Thematik bislang kaum betrachtet worden ist, hat diese Arbeit als ersten wichtigen Beitrag die Anforderungen an eine robuste Ausführung mobiler Aktivitäten in einer Prozessumgebung abgeleitet. Dazu wurden Fallstudien und eine Literaturstudie durchgeführt. Des Weiteren wurden mobile Prototypen entwickelt, um die Anforderungserhebung aus fachlicher Sicht zu ermöglichen. Weiters wurden existierende Ansätze, mobile Anwendungen zu entwickeln, mittels der Prototypen untersucht. Letzteres ist ein wesentlicher Aspekt, wenn Smart-Mobilgeräte in eine Prozessumgebung integriert werden sollen. Schließlich resultierte aus der Anforderungserhebung ein detaillierter Katalog von Anforderungen, der zur robusten Ausführung mobiler Aktivitäten in einer Prozessumgebung adressiert werden muss.

Der Kernbeitrag dieser Arbeit ist ein Rahmenwerk, das den abgeleiteten Anforderungskatalog erfüllt. Das Rahmenwerk fußt auf **fünf** technischen Säulen, die aufeinander aufbauen:

1. Mobiler Kontext
2. Modellierung, Erzeugung und Ausführung mobiler Aktivitäten
3. Mobile Aktivitäten und Ausnahmebehandlung
4. Mobile Aktivitäten und Prozess-Constraints
5. Architekturaspekte

Zu allen Säulen wurden technische Lösungen vorgestellt. Die technische Lösung der ersten Säule besteht aus einem umfangreichen Parameterkatalog, der den Kontext einer mobilen Aktivität spezifisch erfasst. So werden zum Beispiel mobile Parameter erfasst, die eine Offline-Ausführung ermöglichen oder bestimmen, wie nahe sich ein mobiler Endanwender am Ausführungsort der mobilen Aktivität aufhält. Die technischen Lösungen der Säulen 2-5 werden im Kern durch zwei Maßnahmen realisiert, die wiederum aufeinander aufbauen. Es werden einerseits Algorithmen vorgestellt, die den Parameterkatalog der ersten Säule bewerten (Maßnahme 1). Andererseits werden Konzepte eingeführt, welche die berechneten Ergebnisse der Algorithmen geeignet in eine Prozessumgebung integrieren (Maßnahme 2). So muss in Säule 2 zur Realisierung einer Offline-Ausführung (Ergebnis Maßnahme 1) ein neues Protokoll (Ergebnis Maßnahme 2) zwischen Smart-Mobilgerät und PrMS entwickelt werden.

Die *fünf* Säulen wurden zudem so konzipiert, dass die Konzepte der einzelnen Säulen aufeinander aufbauen. So berücksichtigen die Konzepte zu den Prozess-Constraints zum Beispiel die Konzepte zur Ausnahmebehandlung. Liegt ein Prozess-Constraint für eine mobile Aktivität vor und wird eine Ausnahmebehandlung für diese Aktivität notwendig, ist die Ausnahmebehandlung weiterhin ohne Einschränkungen möglich. Ohne die Integration der *fünf* Säulen hätte der Anforderungskatalog nicht vollständig abgedeckt werden können. Die Integration stellt daher einen wesentlichen Beitrag der Arbeit dar.

Schließlich wurde ein Prototyp, der in verschiedenen Fallstudien eingesetzt wurde, entwickelt. Es konnte einerseits gezeigt werden, dass er anwendbar ist, andererseits dass sich die Konzepte des Rahmenwerks in der Praxis bewähren.

Zusammengefasst liefert das in dieser Dissertation vorgestellte Rahmenwerk zur robusten Ausführung mobiler Aktivitäten einen wichtigen Forschungsbeitrag zur Prozess-Management-Technologie. Vor allem die integrierte Sicht des Rahmenwerks liefert für den Forschungsbereich neue Beiträge. Abbildung 14.1 fasst die Ergebnisse der Dissertation grafisch zusammen.

Abbildung 14.1 illustriert darüber hinaus, dass das Rahmenwerk bis auf das Monitoring [PAC⁺07, LR13b] den kompletten Lebenszyklus mobiler Aktivitäten unterstützt. Durch die Fokussierung auf die robuste Ausführung mobiler Aktivitäten wurde dieser Aspekt ausgeklammert. Wird das Monitoring zusätzlich berücksichtigt, sollten Portal-Techniken [Eck10, NKK08] einbezogen werden. Diese versprechen im Zusammenhang mit mobilen Aktivitäten gute Visualisierungsmöglichkeiten, da ein mobiler Zugriff auf die Visualisierung durch den Einsatz von Web-Techniken möglich ist. Darüber hinaus sollten Sichten-Konzepte und Abstraktionen für komplexe Prozesse [KR13b, KKR12a, KKR12b, KR13a, KR13c, RKBB12] integriert werden. So könnten durch Berücksichtigung der Parameter des mobilen Kontexts zum Beispiel kontextbezogene Prozesssichten nützlich sein.

Durch die intensive Auseinandersetzung mit mobilen Aktivitäten sowie den durchgeführten Fallstudien wurden weitere Fragestellungen über das Monitoring hinausgehend aufgeworfen, die Themen für zukünftige Arbeiten liefern.

Da mobile Aktivitäten auch zu neuen Einträgen in der Ausführungshistorie führen, ermöglichen Mining-Techniken [LRW09b, LRW09a, LRW10, LRW11, Aal11, AWM04, ADH⁺03] eine Analyse, ob aus diesen Einträgen neue Erkenntnisse gewonnen werden können. In den Fallstudien im Gesundheitswesen wurde dieser Aspekt zudem von den Fachanwendern öfters nachgefragt.

In verschiedenen Fallstudien kam die weitere Anforderung von Endanwendern, dass diese gerne ein Endanwenderprofil pflegen würden [CFZ01]. Ein solches Profil könnte ausdrücken, an welchen mobilen Aktivitäten man besonders interessiert ist bzw. welche man für die Ausführung

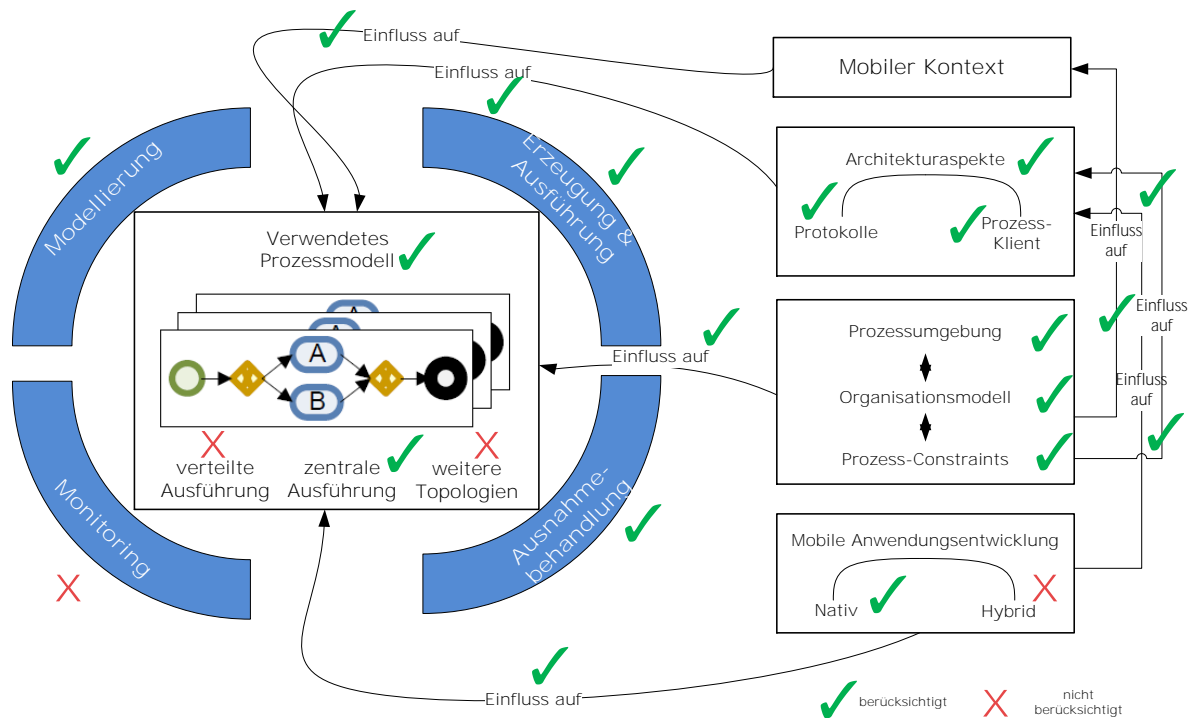


Abbildung 14.1: Rahmenwerk zur robusten Ausführung mobiler Aktivitäten in Prozessumgebungen

ausschließen möchte. Im Zusammenhang von Profilen spielen auch Aspekte wissensintensiver Prozesse [MR14, MKR12, PMLR14] eine Rolle, die zur Erstellung von Profilen genutzt werden können.

Die vorliegende Arbeit setzt eine aktivitätenszentrierte Beschreibung von Prozessen voraus. Die Fallstudien haben allerdings auch gezeigt, dass im Zusammenhang mobiler Aktivitäten viele Daten gelesen und geschrieben werden, die zudem dauerhaft gespeichert werden müssen. Daher sollten Konzepte untersucht werden, die sich mit einer engen Integration von Daten und Prozessen (z.B. Objekt-zentrierte oder Daten-getriebene Prozesse) beschäftigen [KR11, Kü13, CKR13a, CKR13b, CKAR14, CKR14, CRRCA11]. Weiters sollten auch Ansätze mit dem Fokus einer mobilen Datenerfassung [WCK11, SSP⁺14, KWM⁺08, CGG⁺05, Sin07] untersucht werden.

Zusätzlich spielen gesetzliche Auflagen im Kontext mobiler Aktivitäten eine Rolle. Daher sollte die Anwendung von Compliance-Konzepten [Ly13, LRMGD09, CRR11b] auf Szenarien mit mobilen Aktivitäten untersucht werden. Dies wird in einer verteilten Umgebung mit Organisationsübergreifenden Prozessen noch wichtiger [KRL⁺13, SKR14, CRR10].

Der Parameter für die Dringlichkeit hat gezeigt, dass Zeitaspekte für mobile Aktivitäten wichtig sind. Ihre Beherrschbarkeit im mobilen Anwendungskontext wirft jedoch viele Herausforderungen auf (siehe [LWR14, LR14a, LR14b, LPCR13, LPCR15, LKR13, UELW10] für entsprechende Arbeiten zur Zeitperspektive in Prozessen).

Der mobile Kontext wird durch Parameter beschrieben, die quantitative Bewertungen ermöglichen. In diesem Zusammenhang wurden in den Fallstudien von Endanwendern auch Funktionen zur Wirtschaftlichkeitsbetrachtung gefordert [RB90a, RB90b]. Ferner ist zu

untersuchen, inwieweit eine mobile Aktivitätenunterstützung am Ort der Tätigkeit die Prozess-Qualität [LR15, LR13b, LR13a, LR12] verbessert.

Literaturverzeichnis

- [Aal11] AALST, W.M.P. van d.: *Process Mining: Discovery, Conformance and Enhancement of Business Processes*. Springer Science & Business Media, 2011
- [ABBH00] AMMENWERTH, E. ; BUCHAUER, A. ; BLUDAU, B. ; HAUX, R.: Mobile Information and Communication Tools in the Hospital. In: *Int'l Journal of Medical Informatics* 57 (2000), Nr. 1, S. 21–40
- [ACKM04] ALONSO, G. ; CASATI, F. ; KUNO, H. ; MACHIRAJU, V.: *Web Services*. Springer, 2004
- [ADH⁺03] AALST, W.M.P. van d. ; DONGEN, B. van ; HERBST, J. ; MARUSTER, L. ; SCHIMM, G. ; WEIJTERS, A.: Workflow Mining: A Survey of Issues and Approaches. In: *Data & Knowledge Engineering* 47 (2003), Nr. 2, S. 237–267
- [AGA⁺95] ALONSO, G. ; GÜNTHÖR, R. ; AGRAWAL, D. ; EL ABBADI, A. ; KAMATH, M. ; MOHAN, C.: Exotica/FMDC: Handling Disconnected Clients in a Workflow Management System. In: *Proc 3rd Int'l Conf on Cooperative Systems* (1995), S. 99–110
- [All08] ALLWEYER, T.: Vom fachlichen Modell zum ausführbaren Workflow. Am Beispiel von ARIS und Intalio|BPMS. In: *Fachhochschule Kaiserslautern* (2008)
- [And14] *Android*. [http://de.wikipedia.org/wiki/Android_\(Betriebssystem\)](http://de.wikipedia.org/wiki/Android_(Betriebssystem)), 2014. – [Online; accessed 19-June-2015]
- [APP14a] *App Store*. http://de.wikipedia.org/wiki/App_Store, 2014. – [Online; accessed 19-June-2015]
- [App14b] *Appcelerator*. <http://en.wikipedia.org/wiki/Appcelerator>, 2014. – [Online; accessed 19-June-2015]
- [Ara14] ARAS, A.: Design und Konzeption einer mobilen Anwendung zur Unterstützung tinnitusgeschädigter Patienten. In: *Bachelor Thesis, Ulm University* (2014)
- [AVH04] AALST, W.M.P. van d. ; VAN HEE, K.: *Workflow Management: Models, Methods, and Systems*. MIT Press, 2004
- [AW05] ATLURI, V. ; WARNER, J.: Supporting Conditional Delegation in Secure Workflow Management Systems. In: *Proc 10th ACM Symposium on Access Control Models and Technologies* ACM, 2005, S. 49–58
- [AWM04] AALST, W.M.P. van d. ; WEIJTERS, T. ; MARUSTER, L.: Workflow Mining: Discovering Process Models from Event Logs. In: *IEEE Transactions on Knowledge and Data Engineering* 16 (2004), Nr. 9, S. 1128–1142
- [Bau01] BAUER, T.: Effiziente Realisierung unternehmensweiter Workflow-Management-Systeme. In: *PhD Thesis, Ulm University* (2001)
- [BBG⁺95] BERENSON, H. ; BERNSTEIN, P. ; GRAY, J. ; MELTON, J. ; O'NEIL, E. ; O'NEIL, P.: A Critique of ANSI SQL Isolation Levels. In: *ACM SIGMOD Record* Bd. 24 ACM, 1995, S. 1–10

- [BBG06] BOHN, H. ; BOBEK, A. ; GOLATOWSKI, F.: SIRENA-Service Infrastructure for Real-time Embedded Networked Devices: A Service Oriented Framework for Different Domains. In: *Int'l Conf on Mobile Communications and Learning Technologies* IEEE, 2006, S. 43–43
- [BBP09] BUCHWALD, S. ; BAUER, T. ; PRYSS, R.: IT-Infrastrukturen für flexible, service-orientierte Anwendungen - ein Rahmenwerk zur Bewertung. In: *13 GI-Fachtagung Datenbanksysteme für Business, Technologie und Web*, Koellen-Verlag, 2009 (Lecture Notes in Informatics P-144), S. 524–543
- [BBR11] BUCHWALD, S. ; BAUER, T. ; REICHERT, M.: Bridging the Gap Between Business Process Models and Service Composition Specifications. In: *Service Life Cycle Tools and Technologies: Methods, Trends and Advances* (2011), S. 124–153
- [BBR13] BAUER, T. ; BUCHWALD, S. ; REICHERT, M.: Improving the Quality and Cost-effectiveness of Process-oriented, Service-driven Applications: Techniques for Enriching Business Process Models. In: *Service-Driven Approaches to Architecture and Enterprise Integration* (2013), S. 104–134
- [BBTR14] BAUER, T. ; BUCHWALD, S. ; TIEDEKEN, J. ; REICHERT, M.: Konzeption eines SOA-Repository mit Analysefähigkeiten. In: *Workshop IT-Governance und Strategisches Informationsmanagement* (2014)
- [BCC⁺99] BIERE, A. ; CIMATTI, A. ; CLARKE, E. ; FUJITA, M. ; ZHU, Y.: Symbolic Model Checking Using SAT Procedures Instead of BDDs. In: *Proc 36th Annual IEEE Design Automation Conf* ACM, 1999, S. 317–320
- [BD00a] BAUER, T. ; DADAM, P.: Efficient Distributed Workflow Management Based on Variable Server Assignments. In: *Proc. 12th Int'l Conf on Advanced Information Systems Engineering*, Springer, 2000 (LNCS 1789), S. 94–109
- [BD00b] BAUER, T. ; DADAM, P.: Variable Serverzuordnungen und komplexe Bearbeiterzuordnungen im Workflow-Management-System ADEPT. In: *Technical Report* (2000), Nr. UIB-2000-02
- [BD05] BECKER, C. ; DÜRR, F.: On Location Models for Ubiquitous Computing. In: *Personal Ubiquitous Computing* 9 (2005), Nr. 1, S. 20–31. – ISSN 1617–4909
- [BFA97] BERTINO, E. ; FERRARI, E. ; ATLURI, V.: A Flexible Model Supporting the Specification and Enforcement of Role-based Authorization in Workflow Management Systems. In: *Proc 2nd ACM Workshop on Role-based Access Control* ACM, 1997, S. 1–12
- [BFA99] BERTINO, E. ; FERRARI, E. ; ATLURI, V.: The Specification and Enforcement of Authorization Constraints in Workflow Management Systems. In: *ACM Transactions on Information and System Security* 2 (1999), Nr. 1, S. 65–104
- [BGF⁺09] BATTISTA, D. ; GRAZIANO, D. ; FRANCHI, V. ; RUSSO, A. ; LEONI, M. de ; MECELLA, M.: A Web Service-Based Process-Aware Information System for Smart Devices. In: *Technical Report, University of Rome, Department of Computer and System Sciences* 1 (2009), Nr. 5

- [BMR96] BARBARA, D. ; MEHROTRA, S. ; RUSINKIEWICZ, M.: INCAs: Managing Dynamic Workflows in Distributed Environments. In: *Journal of Database Management* 7 (1996), Nr. 1, S. 5–15
- [BPE14a] *Web Services Business Process Execution Language Version 2.0*. <http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.pdf>, 2014. – [Online; accessed 19-June-2015]
- [BPE14b] *WS-BPEL Extension for People (BPEL4People) Specification Version 1.1*. <http://docs.oasis-open.org/bpel4people/bpel4people-1.1-spec-cs-01.pdf>, 2014. – [Online; accessed 19-June-2015]
- [BPM14a] *Business Process Model and Notation*. <http://www.omg.org/spec/BPMN/2.0.2/>, 2014. – [Online; accessed 19-June-2015]
- [BPM14b] *Business Process Model and Notation ISO Release Information*. <http://www.omg.org/spec/BPMN/ISO/19510/PDF>, 2014. – [Online; accessed 19-June-2015]
- [BTM14a] *Betäubungsmittel-Verschreibungsverordnung*. <http://de.wikipedia.org/wiki/Betäubungsmittel-Verschreibungsverordnung>, 2014. – [Online; accessed 19-June-2015]
- [BTM14b] *Betäubungsmittelgesetz Deutschland*. [https://de.wikipedia.org/wiki/Betäubungsmittelgesetz_\(Deutschland\)](https://de.wikipedia.org/wiki/Betäubungsmittelgesetz_(Deutschland)), 2014. – [Online; accessed 19-June-2015]
- [BTR10] BREITSCHWERDT, R. ; THOMAS, O. ; ROBERT, S.n: Mobile Anwendungssysteme zur Unterstützung Ambulanter Pflegedienstleistungen: Anforderungsanalyse und Einsatzpotenziale. In: *Proc MoCoMed* (2010)
- [Buc12] BUCHWALD, S.: Erhöhung der Flexibilität und Durchgängigkeit prozessorientierter Applikationen mittels Service-Orientierung. In: *PhD Thesis, Ulm University* (2012)
- [BVA06] BREAU, T. ; VAIL, M. ; ANTON, A.: Towards Regulatory Compliance: Extracting Rights and Obligations to Align Requirements with Regulations. In: *14th IEEE Int'l Conf on Requirements Engineering* IEEE, 2006, S. 49–58
- [BW13] BYEONG-WAN, K.: Phablet Phenom: Samsung Galaxy Note Pens Best-Seller. In: *SERI Quarterly* 6 (2013), Nr. 2
- [CBRZ01] CLARKE, E. ; BIERE, A. ; RAIMI, R. ; ZHU, Y.: Bounded Model Checking Using Satisfiability Solving. In: *Formal Methods in System Design* 19 (2001), Nr. 1, S. 7–34
- [CEM01a] CAPRA, L. ; EMMERICH, W. ; MASCOLO, C.: Middleware for Mobile Computing: Awareness vs. Transparency. In: *Proc 8th Workshop on Hot Topics in Operating System* (2001)
- [CEM01b] CAPRA, L. ; EMMERICH, W. ; MASCOLO, C.: Reflective Middleware Solutions for Context-Aware Applications. In: *Metalevel Architectures and Separation of Crosscutting Concerns*. Springer, 2001, S. 126–133
- [CF14] *Cross Frameworks*. http://en.wikipedia.org/wiki/Multiple_phone_web-based_application_framework, 2014. – [Online; accessed 19-June-2015]

- [CFL03] CABRI, G. ; FERRARI, L. ; LEONARDI, L.: Enabling Mobile Agents to Dynamically Assume Roles. In: *Proc ACM Symposium on Applied Computing* ACM, 2003, S. 56–60
- [CFL04] CABRI, G. ; FERRARI, L. ; LEONARDI, L.: Agent Role-based Collaboration and Coordination: A Survey About Existing Approaches. In: *IEEE Int'l Conf on Systems, Man and Cybernetics* Bd. 6 IEEE, 2004, S. 5473–5478
- [CFZ01] CHERNIACK, M. ; FRANKLIN, M. ; ZDONIK, S.: Expressing User Profiles for Data Recharging. In: *IEEE Personal Communications* 8 (2001), Nr. 4, S. 32–38
- [CGG⁺05] CURINO, C. ; GIANI, M. ; GIORGETTA, M. ; GIUSTI, A. ; MURPHY, A. ; PICCO, G.: Mobile Data Collection in Sensor Networks: The TinyLime Middleware. In: *Pervasive and Mobile Computing* 1 (2005), Nr. 4, S. 446–469
- [CGP99] CLARKE, E. ; GRUMBERG, O. ; PELED, D.: *Model Checking*. MIT Press, 1999
- [CGR⁺13] CABANILLAS, C. ; GARCIA, J. ; RESINAS, M. ; RUIZ, D. ; MENDLING, J. ; RUIZ-CORTES, A.: Priority-Based Human Resource Allocation in Business Processes. In: *Proc 11th Int'l Conf on Service-Oriented Computing*. Springer, 2013, S. 374–388
- [CK00] CHEN, G. ; KOTZ, D.: A Survey of Context-Aware Mobile Computing Research / Dept of Computer Science Dartmouth College. 2000 (TR2000-381). – Forschungsbericht
- [CK08a] CRAMPTON, J. ; KHAMBHAMMETTU, H.: Delegation and Satisfiability in Workflow Systems. In: *Proc 13th ACM Symposium on Access Control Models and Technologies* ACM, 2008, S. 31–40
- [CK08b] CRAMPTON, J. ; KHAMBHAMMETTU, H.: Delegation in Role-based Access Control. In: *Int'l Journal of Information Security* 7 (2008), Nr. 2, S. 123–136
- [CK08c] CRAMPTON, J. ; KHAMBHAMMETTU, H.: On Delegation and Workflow Execution Models. In: *Proc 2008 ACM Symposium on Applied Computing* ACM, 2008, S. 2137–2144
- [CKAR14] CHIAO, C. ; KÜNZLE, V ; ANDREWS, K. ; REICHERT, M.: A Tool for Supporting Object-Aware Processes. In: *Proc Workshops 18th Int'l Conf on Distributed Object Computing*, IEEE Computer Society Press, 2014, S. 410–413
- [CKR13a] CHIAO, C. ; KÜNZLE, V ; REICHERT, M.: Integrated Modeling of Process- and Data-Centric Software Systems with PHILharmonicFlows. In: *1st Int'l Workshop on Communicating Business Process and Software Models*, IEEE Computer Society Press, 2013, S. 1–10
- [CKR13b] CHIAO, C. ; KÜNZLE, V ; REICHERT, M.: Object-Aware Process Support in Healthcare Information Systems: Requirements, Conceptual Framework and Examples. In: *Int'l Journal on Advances in Life Sciences* 5 (2013), Nr. 1 & 2, S. 11–26
- [CKR14] CHIAO, C. ; KÜNZLE, V ; REICHERT, M.: Towards Schema Evolution in Object-Aware Process Management Systems. In: *Int'l Workshop on the Evolution of Information Systems and their Design Methods*, Koellen-Verlag, 2014 (LNCS P-234), S. 101–115

- [CKR⁺15] CABANILLAS, C. ; KNUPLESCH, D. ; RESINAS, M. ; REICHERT, M. ; MENDLING, J. ; RUIZ-CORTES, A.: RALph: A Graphical Notation for Resource Assignments in Business Processes. In: *27th Int'l Conf on Advanced Information Systems Engineering*, Springer, June 2015 (LNCS 9097), S. 53–68
- [CLK99] CHIU, D. ; LI, Q. ; KARLAPALEM, K.: A Meta Modeling Approach to Workflow Management Systems Supporting Exception Handling. In: *Information Systems* 24 (1999), Nr. 2, S. 159–184
- [CNB⁺13] CROMBACH, A. ; NANDI, C. ; BAMBONYE, M. ; LIEBRECHT, M. ; PRYSS, R. ; REICHERT, M. ; ELBERT, T. ; WEIERSTALL, R.: Screening for Mental Disorders in Post-Conflict Regions Using Computer Apps - A Feasibility Study from Burundi. In: *XIII Congress of European Society of Traumatic Stress Studies*, 2013, S. 70–70
- [Cor14] *Apache Cordova*. <http://cordova.apache.org/>, 2014. – [Online; accessed 19-June-2015]
- [Cov72] COVER, T.: Broadcast Channels. In: *IEEE Transactions on Information Theory* 18 (1972), Nr. 1, S. 2–14
- [Cra03] CRAMPTON, J.: Specifying and Enforcing Constraints in Role-based Access Control. In: *Proc 8th ACM Symposium on Access Control Models and Technologies* ACM, 2003, S. 43–50
- [Cra04] CRAMPTON, J.: An Algebraic Approach to the Analysis of Constrained Workflow Systems. In: *Proc 3rd Workshop on Foundations of Computer Security* Citeseer, 2004, S. 61–74
- [CRE14] *Credentials*. <http://en.wikipedia.org/wiki/Credential>, 2014. – [Online; accessed 19-June-2015]
- [CRORC12] CABANILLAS, C. ; RIO-ORTEGA, A. del ; RESINAS, M. ; CORTES, A.: CRISTAL: Collection of Resource-centRiC Supporting Tools And Languages. In: *Proc BPM 2012 Demos*. Springer, 2012, S. 51–56
- [CRRC10] CABANILLAS, C. ; RESINAS, M. ; RUIZ-CORTES, A.: Hints On How to Face Business Process Compliance. In: *Actas de los Talleres de las Jornadas de Ingenieria del Software y Bases de Datos 4* (2010), Nr. 4, S. 26–32
- [CRRC11a] CABANILLAS, C. ; RESINAS, M. ; RUIZ-CORTES, A.: Defining and Analysing Resource Assignments in Business Processes with RAL. In: *Proc 9th Int'l Conf on Service-Oriented Computing*. Springer, 2011, S. 477–486
- [CRRC11b] CABANILLAS, C. ; RESINAS, M. ; RUIZ-CORTES, A.: Exploring Features of a Full-Coverage Integrated Solution for Business Process Compliance. In: *Advanced Information Systems Engineering Workshops* Springer, 2011, S. 218–227
- [CRRC12a] CABANILLAS, C. ; RESINAS, M. ; RUIZ-CORTÉS, A.: Automated Resource Assignment in BPMN Models using RACI Matrices. In: *Proc On the Move to Meaningful Internet Systems*. Springer, 2012, S. 56–73
- [CRRC12b] CABANILLAS, C. ; RESINAS, M. ; RUIZ-CORTÉS, A.: RAL: A High-Level User-Oriented Resource Assignment Language for Business Processes. In: *Proc BPM Workshops* Springer, 2012, S. 50–61

- [CRRC13] CABANILLAS, C. ; RESINAS, M. ; RUIZ-CORTES, A.: Designing Business Processes with History-Aware Resource Assignments. In: *Proc BPM 2012 Workshops*. Springer, 2013, S. 101–112
- [CRRCA11] CABANILLAS, C. ; RESINAS, M. ; RUIZ-CORTES, A. ; AWAD, A.: Automatic Generation of a Data-Centered View of Business Processes. In: *Advanced Information Systems Engineering* Springer, 2011, S. 352–366
- [Cum07] CUMBERLIDGE, M.: *Business Process Management with JBoss jBPM*. Packt Publishing Ltd, 2007
- [Dö14] DÖTZL, V.: Design und Konzeption einer Applikation zur Unterstützung tinnitusgeschädigter Patienten. In: *Bachelor Thesis, Ulm University* (2014)
- [DAS01] DEY, A. ; ABOWD, G. ; SALBER, D.: A Conceptual Framework and A Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications. In: *Human-Computer Interaction* 16 (2001), Nr. 2, S. 97–166
- [Dea14] *Deadlock*. <http://de.wikipedia.org/wiki/Deadlock>, 2014. – [Online; accessed 19-June-2015]
- [Dec11] DECKER, M.: Modellierung von Ortseinschränkungen für mobile Geschäftsprozesse mit höheren Petri-Netzen. In: *MMS*, 2011, S. 105–118
- [DEM14] *DEMAC*. <https://vsis-www.informatik.uni-hamburg.de/vsys/research/lookproject/29>, 2014. – [Online; accessed 19-June-2015]
- [DLNW13] DINH, H. ; LEE, C. ; NIYATO, D. ; WANG, P.: A Survey of Mobile Cloud Computing: Architecture, Applications, and Approaches. In: *Wireless Communications and Mobile Computing* 13 (2013), Nr. 18, S. 1587–1611
- [DLRMR13] DUMAS, M. ; LA ROSA, M. ; MENDLING, J. ; REIJERS, H.A.: *Fundamentals of Business Process Management*. Springer, 2013
- [DPS+94] DEMERS, A. ; PETERSEN, K. ; SPREITZER, M. ; FERRY, D. ; THEIMER, M. ; WELCH, B.: The Bayou Architecture: Support for Data Sharing Among Mobile Users. In: *Proc of 1st Workshops on Mobile Computing Systems and Applications* IEEE, 1994, S. 2–7
- [DRRM+09a] DADAM, P. ; REICHERT, M. ; RINDERLE-MA, S. ; GOESER, K. ; KREHER, U. ; JURISCH, M.: Von ADEPT zur AristaFlow BPM Suite - Eine Vision wird Realität: Correctness by Construction und flexible, robuste Ausführung von Unternehmensprozessen. In: *EMISA Forum* 29 (2009), Nr. 1, S. 9–28
- [DRRM+09b] DADAM, P. ; REICHERT, M. ; RINDERLE-MA, S. ; LANZ, A. ; PRYSS, R. ; PREDESCHLY, M. ; KOLB, J. ; LY, T. ; JURISCH, M. ; KREHER, U. ; GOESER, K.: From ADEPT to AristaFlow BPM Suite: A Research Vision has become Reality. In: *Proc 1st Int'l Workshop on Empirical Research in Business Process Management*, Springer, 2009 (LNBIP 43), S. 529–531
- [DVG10] DIJKMAN, R. ; VAN GORP, P.: BPMN 2.0 Execution Semantics Formalized As Graph Rewrite Rules. In: *Business Process Modeling Notation*. Springer, 2010, S. 16–30

- [Eck10] ECKERSON, W.: *Performance Dashboards: Measuring, Monitoring, and Managing Your Business*. John Wiley & Sons, 2010
- [EDG14] *Enhanced Data Rates for GSM Evolution*. http://de.wikipedia.org/wiki/Enhanced_Data_Rates_for_GSM_Evolution, 2014. – [Online; accessed 19-June-2015]
- [Eic12] EICHWALD, A.: Einsatz von Prozess-Management-Technologie in der Klinischen Visite mithilfe der AristaFlow BPM Suite. In: *Bachelor Thesis, Ulm University* (2012)
- [Ene11] ENENKIEL, K.: healthHistory - Konzeption & Implementierung einer mobilen Patientenakte. In: *Diploma Thesis, Ulm University* (2011)
- [Erl04] ERL, T.: *Service-Oriented Architecture: A Field Guide to Integrating XML and Web Services*. Prentice Hall PTR, 2004
- [Erl05] ERL, T.: *Service-Oriented Architecture (SOA) Concepts, Technology and Design*. Prentice Hall PTR, 2005
- [Fab12] FABIAN, M.: Entwicklung eines mobilen und Service getriebenen Workflow-Clients zur Unterstützung von evaluierten Studien der Klinischen Psychologie am Beispiel der AristaFlow BPM Suite und Android. In: *Bachelor Thesis, Ulm University* (2012)
- [FGK⁺05] FOSTER, N. ; GREENWALD, M. ; KIRKEGAARD, C. ; PIERCE, B. ; SCHMITT, A.: Exploiting Schemas in Data Synchronization. In: *Database Programming Languages* Springer, 2005, S. 42–57
- [FGL⁺96] FOURNET, C. ; GONTHIER, G.s ; LEVY, J. ; MARANGET, L. ; REMY, D.: A Calculus of Mobile Agents. In: *Proc of Concurrency Theory*. Springer, 1996, S. 406–421
- [FGP⁺03] FENG, X. ; GUTIERREZ, J. ; PRATT, M. ; ESLINGER, M. ; FARKASH, N.: Using Model Checking to Prove Constraints of Combinational Equivalence Checking. In: *Proc Conf on Design Verification*, 2003
- [For14] *Formfaktor*. [http://de.wikipedia.org/wiki/Formfaktor_\(Computertechnik\)](http://de.wikipedia.org/wiki/Formfaktor_(Computertechnik)), 2014. – [Online; accessed 19-June-2015]
- [FPO14] *Floating Point Operations Per Second*. http://de.wikipedia.org/wiki/Floating_Point_Operations_Per_Second, 2014. – [Online; accessed 19-June-2015]
- [Gao14] GAO, Y.: *BPMN-BPEL Transformation and Round Trip Engineering*. <http://www.eclarus.com/pdf/BPMNBPELMapping.pdf>, 2014. – [Online; accessed 19-June-2015]
- [GAR14a] *BYOD & Enterprise Mobility*. <http://www.gartner.com/technology/topics/mobile.jsp>, 2014. – [Online; accessed 19-June-2015]
- [GAR14b] *Gartner Inc*. http://de.wikipedia.org/wiki/Gartner_Inc., 2014. – [Online; accessed 19-June-2015]

- [GAR14c] *Service-Oriented Architecture Scenario*. <https://www.gartner.com/doc/391595/serviceoriented-architecture-scenario>, 2014. – [Online; accessed 19-June-2015]
- [GAR14d] *The Mobile Imperative: Getting Started and Overcoming Obstacles*. <https://www.gartner.com/doc/2625818/getting-started-overcoming-obstacles-mobile>, 2014. – [Online; accessed 19-June-2015]
- [GCFP10] GINER, P. ; CETINA, C. ; FONS, J. ; PELECHANO, V.: Developing Mobile Workflow Support in the Internet of Things. In: *IEEE Pervasive Computing* 9 (2010), Nr. 2, S. 18–26
- [Gei14] GEIGER, P.: Optimierung von Geschäftsprozessen mithilfe mobiler Endgeräte und unter Berücksichtigung der zur Verfügung stehenden Sensoren. In: *Master Thesis, Ulm University* (2014)
- [GG99] GEHRING, H. ; GADATSCH, A.: Eine Rahmenarchitektur für Workflow-Management-Systeme. In: *Fachbereichsbericht Nr. 275* (1999)
- [GGGO99] GAVALAS, D. ; GREENWOOD, D. ; GHANBARI, M. ; O'MAHONY, M.: An Infrastructure for Distributed and Dynamic Network Management Based on Mobile Agent Technology. In: *IEEE Int'l Conf on Communications* Bd. 2 IEEE, 1999, S. 1362–1366
- [GHOS96] GRAY, J. ; HELLAND, P. ; O'NEIL, P. ; SHASHA, D.: The Dangers of Replication and a Solution. In: *ACM SIGMOD Record* Bd. 25 ACM, 1996, S. 173–182
- [Goo14] *Google*. <http://www.google.com/>, 2014. – [Online; accessed 19-June-2015]
- [GOR11] GRAMBOW, G. ; OBERHAUSER, R. ; REICHERT, M.: Event-driven Exception Handling for Software Engineering Processes. In: *5th Int'l Workshop on Event-driven Business Process Management*, Springer, 2011 (LNBIP 99), S. 414–426
- [GPS14] *Global Positioning System*. <http://www.gps.gov/technical/icwg/IS-GPS-200G.pdf>, 2014. – [Online; accessed 19-June-2015]
- [GPSR13] GEIGER, P. ; PRYSS, R. ; SCHICKLER, M. ; REICHERT, M.: Engineering an Advanced Location-Based Augmented Reality Engine for Smart Mobile Devices. In: *UIB-2013-09*, Ulm University, 2013
- [GS14] *Google Scholar*. http://de.wikipedia.org/wiki/Google_Scholar, 2014. – [Online; accessed 19-June-2015]
- [GScl4] *Google Scholar*. <http://scholar.google.com/>, 2014. – [Online; accessed 19-June-2015]
- [GSFC08] GAALOUL, K. ; SCHAAD, A. ; FLEGEL, U. ; CHAROY, F.: A Secure Task Delegation Model for Workflows. In: *2nd Int'l Conf on Emerging Security Information, Systems and Technologies* IEEE, 2008, S. 10–15
- [GSP⁺14] GEIGER, P. ; SCHICKLER, M. ; PRYSS, R. ; SCHOBEL, J. ; REICHERT, M.: Location-based Mobile Augmented Reality Applications: Challenges, Examples, Lessons Learned. In: *10th Int'l Conf on Web Information Systems and Technologies*, 2014, S. 383–394

- [HA95] HO, J. ; AKYILDIZ, I.: Mobile User Location Update and Paging Under Delay Constraints. In: *Wireless Networks* 1 (1995), Nr. 4, S. 413–425
- [HA99] HUANG, W. ; ATLURI, V.: SecureFlow: A Secure Web-enabled Workflow Management System. In: *Proc 4th ACM Workshop on Role-Based Access Control* ACM, 1999, S. 83–94
- [Haa13] HAASE, C.: Konzeption und Realisierung einer Verteilten Workflow-Steuerung für Mobile Endgeräte. In: *Diploma Thesis, University of Hamburg* (2013)
- [Ham09] HAMANN, K.: Parallele Ausführung von Prozessen auf Mobilien Geräten. In: *Diploma Thesis, University of Hamburg* (2009)
- [HC04] HUNG, P. ; CHIU, D.: Developing Workflow-Based Information Integration With Exception Support in A Web Services Environment. In: *Proc 37th Annual Hawaii Int'l Conf on System Sciences* IEEE, 2004, S. 24–33
- [HCS05] HONG, D. ; CHIU, D. ; SHEN, V.: Requirements Elicitation for the Design of Context-Aware Applications in A Ubiquitous Environment. In: *Proc 7th Int'l Conf on Electronic Commerce* ACM, 2005, S. 590–596
- [Her14] HERRMANN, J.: Konzeption und technische Realisierung eines mobilen Frameworks zur Unterstützung tinnitusgeschädigter Patienten. In: *Diploma Thesis, Ulm University* (2014)
- [HHGR06] HACKMANN, G. ; HAITJEMA, M. ; GILL, C. ; ROMAN, G.: Sliver: A BPEL Workflow Process Execution Engine for Mobile Devices. In: *Proc 4th Int'l Conf on Service Oriented Computing*. Springer, 2006, S. 503–508
- [HMPR04] HEVNER, A. von ; MARCH, S. ; PARK, J. ; RAM, S.: Design Science in Information Systems Research. In: *MIS Quarterly* 28 (2004), Nr. 1, S. 75–105
- [HO03] HANSEN, F. ; OLESHCHUK, V.: SRBAC: A Spatial Role-based Access Control Model for Mobile Systems. In: *Proc 7th Nordic Workshop on Secure IT Systems* Citeseer, 2003, S. 129–141
- [HS07] HOWE, N. ; STRAUSS, W.: *Millennials Go To College*. LifeCourse Associates Great Falls, VA, 2007
- [HS09] HAHN, K. ; SCHWEPPE, H.: Exploring Transactional Service Properties for Mobile Service Composition. In: *Proc 4th Conf Mobile und Ubiquitäre Informationssysteme* 146 (2009), S. 39–52
- [HSS08] HELKIÖ, P. ; SEPPÄLÄ, A. ; SYD, O.: Evaluation of Intalio BPM Tool. In: *Special Course in Information System Integration* (2008)
- [HST07] HANAK, D. ; SZIJARTO, G. ; TAKACS, B.: A Mobile Approach to Ambient Assisted Living. In: *Proc IADIS Multi Conf on Computer Science and Information Systems*, 2007, S. 3–8
- [i4014] *Industrie 4.0*. http://de.wikipedia.org/wiki/Industrie_4.0, 2014. – [Online; accessed 06-March-2014]
- [IAT14] *Informationsarchitektur*. <http://de.wikipedia.org/wiki/Informationsarchitektur>, 2014. – [Online; accessed 19-June-2015]

- [IBM14a] *Geschäftsprozessmanagement (BPM)*. <http://www-03.ibm.com/software/products/de/category/BPM-SOFTWARE>, 2014. – [Online; accessed 19-June-2015]
- [IBM14b] *IBM Developer Works*. http://www.ibm.com/developerworks/websphere/techjournal/1310_amrhein/1310_amrhein.html, 2014. – [Online; accessed 19-June-2015]
- [IBM14c] *IBM Mobile Business Process Management*. http://www.ibm.com/developerworks/websphere/bpmjournal/1202_pacholski/1202_pacholski.html, 2014. – [Online; accessed 19-June-2015]
- [IBM14d] *IBM Worklight*. <http://www-03.ibm.com/software/products/de/worklight-foundation>, 2014. – [Online; accessed 19-June-2015]
- [iOS14] *iOS*. http://de.wikipedia.org/wiki/Apple_iOS, 2014. – [Online; accessed 19-June-2015]
- [IRLP⁺13] ISELE, D ; RUF-LEUSCHNER, M. ; PRYSS, R. ; SCHAUER, M. ; REICHERT, M. ; SCHOBEL, J. ; SCHINDLER, A. ; ELBERT, T.: Detecting Adverse Childhood Experiences with a Little Help from Tablet Computers. In: *XIII Congress of European Society of Traumatic Stress Studies*, 2013, S. 69–70
- [Jab97] JABLONSKI, S.: Architektur von Workflow-Management-Systemen. In: *Informatik Forschung und Entwicklung* 12 (1997), Nr. 2, S. 72–81
- [Jab13] JABLONSKI, M.: Technische Konzeption und Realisierung einer mobilen Anwendung zur Verwaltung von Patientendaten für Ärzte am Beispiel von Android. In: *Diploma Thesis, Ulm University* (2013)
- [Jan07] JANZEN, A.: Position Determination with WLAN and Bluetooth. In: *Universität Freiburg* (2007)
- [JHS⁺99] JING, J. ; HUFF, K. ; SINHA, H. ; HURWITZ, B. ; ROBINSON, B.: Workflow and Application Adaptations in Mobile Environments. In: *Proc 2nd IEEE Workshop on Mobile Computing Systems and Applications* IEEE, 1999, S. 62–69
- [JS98] JABLONSKI, S. ; STEIN, K.: Ein Vorgehensmodell für Workflow-Management-Anwendungen. In: *Vorgehensmodelle für die betriebliche Anwendungsentwicklung*. Springer, 1998, S. 136–151
- [Kü13] KÜNZLE, V.: Object-aware Process Management. In: *Ulm University, PhD Thesis* (2013)
- [KAH⁺12] KOSTA, S. ; AUCINAS, A. ; HUI, P. ; MORTIER, R. ; ZHANG, X.: Thinkair: Dynamic Resource Allocation and Parallel Execution in the Cloud for Mobile Code Offloading. In: *2012 Proc INFOCOM* IEEE, 2012, S. 945–953
- [KB09] KEJRIWA, A. vind ; BEDEKAR, M.: MobiDSL-a Domain Specific Language for Mobile Web Applications: Developing Applications for Mobile Platform Without Web Programming. In: *Proc. dsmforum.org*, 2009
- [KG04a] KÖHLER, A. ; GRUHN, V.: Lösungsansätze für verteilte mobile Geschäftsprozesse. In: *Elektronische Geschäftsprozesse* (2004), S. 243–255

- [KG04b] KÖHLER, A. ; GRUHN, V.: Mobile Process Landscaping am Beispiel von Vertriebsprozessen in der Assekuranz. In: *Workshop Mobile Commerce*, 2004, S. 12–24
- [KGG⁺06] KRUK, S. ; GRZONKOWSKI, S. ; GZELLA, A. ; WORONIECKI, T. ; CHOI, H.: D-FOAF: Distributed Identity Management with Access Rights Delegation. In: *The Semantic Web-ASWC*. Springer, 2006, S. 140–154
- [KH08] KANG, J. ; HAN, K.: A Business Activity Monitoring System Supporting Real-Time Business Performance Management. In: *3rd Int'l Conf on Convergence and Hybrid Information Technology* Bd. 1 IEEE, 2008, S. 473–478
- [KHR12] KOLB, J. ; HÜBNER, P. ; REICHERT, M.: Automatically Generating and Updating User Interface Components in Process-Aware Information Systems. In: *20th Int'l Conf on Cooperative Information Systems*, Springer, 2012 (LNCS 7565), S. 444–454
- [Kir03] KIRN, S.: Ubiquitous Healthcare: The Onkonet Mobile Agents Architecture. In: *Objects, Components, Architectures, Services, and Applications for a Networked World*. Springer, 2003, S. 265–277
- [Kit12] KITTEL, S.: EMedMobile - Integration mobiler Endgeräte in den ärztlichen Hausbesuchsablauf am Beispiel von Android. In: *Diploma Thesis, Ulm University* (2012)
- [KKR12a] KOLB, J. ; KAMMERER, K. ; REICHERT, M.: Updatable Process Views for Adapting Large Process Models: The proView Demonstrator. In: *Demo Track 10th Int'l Conf on Business Process Management*, 2012 (CEUR Workshop Proc 940), S. 6–11
- [KKR12b] KOLB, J. ; KAMMERER, K. ; REICHERT, M.: Updatable Process Views for User-centered Adaption of Large Process Models. In: *10th Int'l Conf on Service Oriented Computing*, Springer, 2012 (LNCS 7636), S. 484–498
- [KL10] KUMAR, K. ; LU, Y.: Cloud Computing for Mobile Users: Can Offloading Computation Save Energy? In: *Computer* 43 (2010), Nr. 4, S. 51–56
- [KLL09] KO, R. ; LEE, S. ; LEE, E.: Business Process Management (BPM) Standards: A Survey. In: *Business Process Management Journal* 15 (2009), Nr. 5, S. 744–791
- [KM03] KOSHUTANSKI, H. ; MASSACCI, F.: An Access Control Framework for Business Processes for Web Services. In: *Proc ACM Workshop on XML Security* ACM, 2003, S. 15–24
- [KM04] KOSHUTANSKI, H. ; MASSACCI, F.: Interactive Access Control for Web Services. In: *Security and Protection in Information Processing Systems*. Springer, 2004, S. 150–166
- [KOK⁺11] KOCUROVA, A. ; OUSSENA, S. ; KOMISARCZUK, P. ; CLARK, T. ; KRAMER, D.: Towards Improved Distributed Collaborative Workflow Management for Mobile Devices. In: *Proc 1st Int'l SIMPDA Symposium* (2011)

- [KOKC12] KOCUROVA, A. ; OUSSENA, S. ; KOMISARCZUK, P. ; CLARK, T.: Context-Aware Content-Centric Collaborative Workflow Management for Mobile Devices. In: *2nd Int'l Conf on Advanced Collaborative Networks, Systems and Applications*, 2012, S. 54–57
- [KOKC13] KOCUROVA, A. ; OUSSENA, S. ; KOMISARCZUK, P. ; CLARK, T.: MobWEL-Mobile Context-Aware Content-Centric Workflow Execution Language. In: *3rd Int'l Conf on Advanced Collaborative Networks, Systems and Applications*, 2013, S. 61–70
- [KPR12a] KNUPLESCH, D. ; PRYSS, R. ; REICHERT, M.: A Formal Framework for Data-Aware Process Interaction Models. Ulm University, 2012 (UIB-2012-06). – Forschungsbericht
- [KPR12b] KNUPLESCH, D. ; PRYSS, R. ; REICHERT, M.: Data-Aware Interaction in Distributed and Collaborative Workflows: Modeling, Semantics, Correctness. In: *8th IEEE Int'l Conf on Collaborative Computing: Networking, Applications and Worksharing*, IEEE Computer Society Press, 2012, S. 223–232
- [KR11] KÜNZLE, V. ; REICHERT, M.: PHILharmonicFlows: Towards a Framework for Object-aware Process Management. In: *Journal of Software Maintenance and Evolution: Research and Practice* 23 (2011), Nr. 4, S. 205–244
- [KR13a] KOLB, J. ; REICHERT, M.: A Flexible Approach for Abstracting and Personalizing Large Business Process Models. In: *Applied Computing Review* 13 (2013), Nr. 1, S. 6–17
- [KR13b] KOLB, J. ; REICHERT, M.: Data Flow Abstractions and Adaptations through Updatable Process Views. In: *28th Symposium on Applied Computing*, ACM Press, 2013, S. 1447–1453
- [KR13c] KOLB, J. ; REICHERT, M.: Supporting Business and IT through Updatable Process Views: The proView Demonstrator. In: *Demo Track 10th Int'l Conf on Service Oriented Computing*, Springer, 2013 (LNCS 7759), S. 460–464
- [Kre14] KREHER, U.: Konzepte, Architektur und Implementierung adaptiver Prozessmanagementsysteme. In: *PhD Thesis, Ulm University* (2014)
- [KRFRM13] KNUPLESCH, D. ; REICHERT, M. ; FDHILA, W. ; RINDERLE-MA, S.: On Enabling Compliance of Cross-Organizational Business Processes. In: *11th Int'l Conf Business Process Management*, Springer, 2013 (LNCS 8094), S. 146–154
- [KRL⁺13] KNUPLESCH, D. ; REICHERT, M. ; LY, T. ; KUMAR, A. ; RINDERLE-MA, S.: Visual Modeling of Business Process Compliance Rules with the Support of Multiple Perspectives. In: *32nd Int'l Conf on Conceptual Modeling* (2013), Nr. 8217, S. 106–120
- [KRP⁺13] KNUPLESCH, D. ; REICHERT, M. ; PRYSS, R. ; FDHILA, W. ; RINDERLE-MA, S.: Ensuring Compliance of Distributed and Collaborative Workflows. In: *9th IEEE Int'l Conf on Collaborative Computing: Networking, Applications and Worksharing* (2013), S. 133–142
- [KS92] KISTLER, J. ; SATYANARAYANAN, M.: Disconnected Operation in the Coda File System. In: *ACM Transactions on Computer Systems* 10 (1992), Nr. 1, S. 3–25

-
- [Kun05] KUNZE, Christian P.: DEMAC: A Distributed Environment for Mobility-Aware Computing. In: *Adjunct Proc 3rd Int'l Conf on Pervasive Computing* Bd. 191, 2005 (Advances in Pervasive Computing), S. 115–121
 - [KWM⁺08] KLUNGSÖYR, J. ; WAKHOLI, P. ; MACLEOD, B. ; ESCUDERO-PASCUAL, A. ; LESH, N.: Open-ROSA, JavaROSA, GloballyMobile—Collaborations around Open Standards for Mobile Applications. In: *Int'l Conf on M4D Mobile Communication Technology for Development*, 2008
 - [KZL06] KUNZE, C. ; ZAPLATA, S. ; LAMERSDORF, W.: Mobile Process Description and Execution. In: *Distributed Applications and Interoperable Systems* Springer, 2006, S. 32–47
 - [KZL07a] KUNZE, C. ; ZAPLATA, S. ; LAMERSDORF, W.: Abstrakte Dienstklassen zur Realisierung Mobiler Prozesse. In: *KiVS 2007* (2007)
 - [KZL07b] KUNZE, C. ; ZAPLATA, S. ; LAMERSDORF, W.: Mobile Processes: Enhancing Cooperation in Distributed Mobile Environments. In: *Journal of Computers* 2 (2007), Nr. 1, S. 1–11
 - [Lac13] LACHENMAYR, L.: Modellierkonzept zur Erfassung wesentlicher Aspekte bei der mobilen Prozessausführung. In: *Master Thesis, Ulm University* (2013)
 - [Lan12] LANGER, D.: MEDo: Mobile Technik und Prozessmanagement zur Optimierung des Aufgabenmanagements im Kontext der Klinischen Visite. In: *Diploma Thesis, Ulm University* (2012)
 - [LCH⁺11] LU, Q. ; CHEN, Y. ; HAITJEMA, M. ; ROMAN, C. ; GILL, C. ; CHEN, G.: Temporal Planning for Co-Design of Host Scheduling and Workflow Allocation in Mobile Environments. In: *Proc of SPARK at ICAPS* (2011)
 - [LCS05] LIU, Y. ; CLARK, J. ; STEPNEY, S.: “Devices Are People Too” - Using Process Patterns to Elicit Security Requirements in Novel Domains: A Ubiquitous Healthcare Example. In: *Security in Pervasive Computing*. Springer, 2005, S. 31–45
 - [LKHK05] LEIMEISTER, J. ; KRUMHOLTZ, H. ; HORSCH, A. ; KUHN, K.: Mobile IT-Systeme im Gesundheitswesen, Mobile Systeme für Patienten. In: *HMD-Praxis der Wirtschaftsinformatik* 41 (2005), Nr. 244, S. 74–85
 - [LKR13] LANZ, A. ; KOLB, J. ; REICHERT, M.: Enabling Personalized Process Schedules with Time-Aware Process Views. In: *2nd Int'l Workshop on Human-Centric Information Systems*, Springer, 2013 (LNBIP), S. 205–216
 - [LKR10] LANZ, A. ; KREHER, U. ; REICHERT, M. ; DADAM, P.: Enabling Process Support for Advanced Applications with the AristaFlow BPM Suite. In: *Proc BPM Demo Track*, 2010 (CEUR Workshop Proc 615)
 - [LKSP02] LANKHORST, M. ; KRANENBURG, H. van ; SALDEN, A. ; PEDDEMORS, A.: Enabling Technology for Personalizing Mobile Services. In: *Proc 35th Annual Hawaii Int'l Conf on System Sciences* IEEE, 2002, S. 1464–1471
 - [LPCR13] LANZ, A. ; POSENATO, R. ; COMBI, C. ; REICHERT, M.: Controllability of Time-Aware Processes at Run Time. In: *21st Int'l Conf on Cooperative Information Systems*, Springer, 2013 (LNCS 8185), S. 39–56

- [LPCR15] LANZ, A. ; POSENATO, R. ; COMBI, C. ; REICHERT, M.: Simple Temporal Networks with Partially Shrinkable Uncertainty. In: *Int'l Conf on Agents and Artificial Intelligence*, 2015
- [LR12] LOHRMANN, M. ; REICHERT, M.: Efficacy-aware Business Process Modeling. In: *20th Int'l Conf on Cooperative Information Systems*, Springer, 2012 (LNCS 7565), S. 38–55
- [LR13a] LOHRMANN, M. ; REICHERT, M.: Demonstrating the Effectiveness of Process Improvement Patterns. In: *Proc BPMDS*, Springer, 2013 (LNBIP 147), S. 230–245
- [LR13b] LOHRMANN, M. ; REICHERT, M.: Understanding Business Process Quality. In: *Business Process Management - Theory and Applications*. Springer, 2013 (Studies in Computational Intelligence 444), S. 41–73
- [LR14a] LANZ, A. ; REICHERT, M.: Dealing with Changes of Time-Aware Processes. In: *12th Int'l Conf on Business Process Management*, Springer, 2014 (LNCS 8659), S. 217–233
- [LR14b] LANZ, A. ; REICHERT, M.: Enabling Time-Aware Process Support with the ATAPIS Toolset. In: *Demo Proc 12th Int'l Conf on Business Process Management*, 2014
- [LR15] LOHRMANN, M. ; REICHERT, M.: Effective Application of Process Improvement Patterns to Business Processes. In: *Software & Systems Modeling* (2015)
- [LRD10a] LANZ, A. ; REICHERT, M. ; DADAM, P.: Making Business Process Implementations Flexible and Robust: Error Handling in the AristaFlow BPM Suite. In: *CAiSE'10 Demos*, 2010
- [LRD10b] LANZ, A. ; REICHERT, M. ; DADAM, P.: Robust and Flexible Error Handling in the AristaFlow BPM Suite. In: *Proc CAiSE'10 Forum, Information Systems Evolution*, Springer, 2010 (LNBIP 72), S. 174–189
- [LRMGD09] LY, T. ; RINDERLE-MA, S. ; GÖSER, K. ; DADAM, P.: On Enabling Integrated Process Compliance with Semantic Constraints in Process Management Systems. In: *Information Systems Frontiers* (2009), S. 1–25
- [LRW09a] LI, C. ; REICHERT, M. ; WOMBACHER, A.: Discovering Reference Models by Mining Process Variants Using A Heuristic Approach. In: *7th Int'l Conf on Business Process Management*, Springer, 2009 (LNCS 5701), S. 344–362
- [LRW09b] LI, C. ; REICHERT, M. ; WOMBACHER, A.: Mining Based on Learning from Process Change Logs. In: *4th Int'l Workshop on Business Process Intelligence*, Springer, 2009 (LNBIP 17), S. 121–133
- [LRW09c] LI, C. ; REICHERT, M. ; WOMBACHER, A.: What are the Problem Makers: Ranking Activities According to their Relevance for Process Changes. In: *7th Int'l Conf on Web Services*, IEEE Computer Society Press, 2009, S. 51–58
- [LRW10] LI, C. ; REICHERT, M. ; WOMBACHER, A.: The MinAdept Clustering Approach for Discovering Reference Process Models out of Process Variants. In: *Int'l Journal of Cooperative Information Systems* 19 (2010), Nr. 3 & 4, S. 159–203

-
- [LRW11] LI, C. ; REICHERT, M. ; WOMBACHER, A.: Mining Business Process Variants: Challenges, Scenarios, Algorithms. In: *Data & Knowledge Engineering* 70 (2011), Nr. 5, S. 409–434
- [LWR14] LANZ, A. ; WEBER, B. ; REICHERT, M.: Time Patterns for Process-Aware Information Systems. In: *Requirements Engineering* 19 (2014), Nr. 2, S. 113–141
- [Ly13] LY, T.: SeaFlows - A Compliance Checking Framework for Supporting the Process Lifecycle. In: *PhD Thesis, Ulm University* (2013)
- [Mar09] MARINELLI, E.: Hyrax: Cloud Computing on Mobile Devices Using MapReduce / DTIC Document. 2009. – Forschungsbericht
- [MAR14] *MARPLE - MANaging Robust mobile Processes in a compLEx world*. <http://www.uni-ulm.de/in/iui-dbis/forschung/projekte/marple.html>, 2014. – [Online; accessed 19-June-2015]
- [MAS14] *Microsoft Academic Search*. <http://academic.research.microsoft.com/>, 2014. – [Online; accessed 19-June-2015]
- [MC114] *Model Checking*. http://de.wikipedia.org/wiki/Model_Checking, 2014. – [Online; accessed 19-June-2015]
- [MCC14] *MCC Meierhofer*. http://www.meierhofer.de/de_home.meierhofer, 2014. – [Online; accessed 19-June-2015]
- [meg14] *Megacode-Training*. <http://de.wikipedia.org/wiki/Megacode-Training>, 2014. – [Online; accessed 19-June-2015]
- [MGO⁺13] MUDANO, A. ; GARY, L. ; OLIVEIRA, A. ; MELTON, M. ; WRIGHT, N. ; CURTIS, J. ; DELZELL, E. ; HARRINGTON, T. ; KILGORE, M. ; LEWIS, C. Elizabeth et al.: Using Tablet Computers Compared to Interactive Voice Response to Improve Subject Recruitment in Osteoporosis Pragmatic Clinical Trials: Feasibility, Satisfaction, and Sample Size. In: *Patient Preference and Adherence* 7 (2013), S. 517
- [MH08] MANS, M. ; HECKL, D.: Steigerung der Wettbewerbsfähigkeit im Baufinanzierungsgeschäft. In: *Bank Praktiker* (2008), S. 362–367
- [MKR12] MUNDBROD, N. ; KOLB, J. ; REICHERT, M.: Towards a System Support of Collaborative Knowledge Work. In: *1st Int'l Workshop on Adaptive Case Management*, Springer, 2012 (LNBIP 132), S. 31–42
- [MM05] MAURINO, A. ; MODAFFERI, S.: Partitioning Rules for Orchestrating Mobile Information Systems. In: *Personal and Ubiquitous Computing* 9 (2005), Nr. 5, S. 291–300
- [MMY05] MAAMAR, Z. ; MOSTEFAOUI, S. ; YAHYAOU, H.: Toward An Agent-Based and Context-Oriented Approach for Web Services Composition. In: *IEEE Transactions on Knowledge and Data Engineering* 17 (2005), Nr. 5, S. 686–697
- [MOL⁺08] MILUZZO, E. ; OAKLEY, J. ; LU, H. ; LANE, N. ; PETERSON, R. ; CAMPBELL, A.: Evaluating the iPhone as a Mobile Platform for People-Centric Sensing Applications. In: *Proc Int'l Conf on UrbanSense* (2008)

- [MQ114] *Millennials (Millennial Generation)*. <http://whatis.techtarget.com/definition/millennials-millennial-generation>, 2014. – [Online; accessed 19-June-2015]
- [MR14] MUNDBROD, N. ; REICHERT, M.: Process-Aware Task Management Support for Knowledge-Intensive Business Processes: Findings, Challenges, Requirements. In: *Proc Workshops 18th Int'l Conf on Distributed Object Computing*, IEEE Computer Society Press, 2014, S. 116–125
- [MS95] MARCH, S. ; SMITH, G.: Design and Natural Science Research on Information Technology. In: *Decision Support Systems* 15 (1995), Nr. 4, S. 251–266
- [Mue04] MUEHLEN, M. zur: Organizational Management in Workflow Applications–Issues and Perspectives. In: *Information Technology and Management* 5 (2004), Nr. 3-4, S. 271–291
- [Mus14] MUSIOL, S.: A Process Engine Independent Architecture Enabling Robust Execution of Mobile Tasks in Business Processes. In: *Master Thesis, Ulm University* (2014)
- [NFN⁺09] NICOLESCU, V. ; FUNK, B. ; NIEMEYER, P. ; HEILER, M. ; WITTGES, H. ; MORANDELL, T. ; VISINTIN, F. ; STEGEMANN, B. ; KIENEGGER, H.: *Praxishandbuch SAP NetWeaver PI-Entwicklung*. Galileo Press, 2009
- [NFS02] NOLL, M. ; FRÖHLICH, D. ; SCHIEBEL, E.: Knowledge Maps of Knowledge Management Tools — Information Visualization with BibTechMon. In: *Practical Aspects of Knowledge Management* Bd. 2569. Springer, 2002, S. 14–27
- [NKK08] NICOLESCU, V. ; KLAPPERT, K. ; KRUMAR, H.: *SAP NetWeaver Portal*. Galileo Press, 2008
- [NoS14] *NoSQL*. <http://de.wikipedia.org/wiki/NoSQL>, 2014. – [Online; accessed 19-June-2015]
- [ODHA08] OUYANG, C. ; DUMAS, M. ; HM, A. ; AALST, W.M.P. van d.: Pattern-Based Translation of BPMN Process Models to BPEL Web Services. In: *Int'l Journal of Web Services Research* 5 (2008), Nr. 1, S. 42–62
- [orm14] *Objektrelationale Abbildung*. http://de.wikipedia.org/wiki/Objektrelationale_Abbildung, 2014. – [Online; accessed 19-June-2015]
- [Ost07] OSTERWEIL, L.: What We Learn From the Study of Ubiquitous Processes. In: *Software Process: Improvement and Practice* 12 (2007), Nr. 5, S. 399–414
- [P2P14] *Peer-to-Peer*. <http://de.wikipedia.org/wiki/Peer-to-Peer>, 2014. – [Online; accessed 19-June-2015]
- [PAC⁺07] POURSHAHID, A. ; AMYOT, D. ; CHEN, P. ; WEISS, M. ; FORSTER, A.: Business Process Monitoring and Alignment: An Approach Based on the User Requirements Notation and Business Intelligence Tools. In: *WER*, 2007, S. 80–91
- [PÁFMO09] PÉREZ, J. ; ÁLVAREZ, J. ; FERNÁNDEZ-MONTES, A. ; ORTEGA, J.: Service-Oriented Device Integration for Ubiquitous Ambient Assisted Living Environments. In: *Distributed Computing, Artificial Intelligence, Bioinformatics, Soft Computing, and Ambient Assisted Living*. Springer, 2009, S. 843–850

- [Par10] PARMENTER, D.: *Key Performance Indicators (KPI): Developing, Implementing, and Using Winning KPIs*. John Wiley & Sons, 2010
- [PB99] PHATAK, S. ; BADRINATH, B.: Data Partitioning for Disconnected Client Server Databases. In: *Proc 1st ACM Int'l Workshops on Data Engineering for Wireless and Mobile Access* ACM, 1999, S. 102–109
- [PCC⁺14] PENG, T. ; CHI, C. ; CHIASERA, A. ; ARMELLIN, G. ; RONCHETTI, M. ; MATTEOTTI, C. ; PARRA, C. ; KASHYTSYA, A.O. ; VARALTA, A.: Business Process Assignment and Execution in Mobile Environments. In: *Int'l Conf on Collaboration Technologies and Systems*, 2014, S. 267–274
- [PG14] *PhoneGap*. <http://de.wikipedia.org/wiki/PhoneGap>, 2014. – [Online; accessed 19-June-2015]
- [PHZ12] PATHAK, A. ; HU, Y. ; ZHANG, M.: Where is the Energy Spent Inside My APP?: Fine Grained Energy Accounting on Smartphones with Eprof. In: *Proc 7th ACM European Conf on Computer Systems* ACM, 2012, S. 29–42
- [Pla14] *Play Framework*. <https://www.playframework.com/documentation/1.2/deployment>, 2014. – [Online; accessed 19-June-2015]
- [PLRH12] PRYSS, R. ; LANGER, D. ; REICHERT, M. ; HALLERBACH, A.: Mobile Task Management for Medical Ward Rounds - The MEDo Approach. In: *1st Int'l Workshop on Adaptive Case Management*, Springer, 2012 (LNBIP 132), S. 43–54
- [PM14] *Prozessmanagement*. <http://de.wikipedia.org/wiki/Prozessmanagement>, 2014. – [Online; accessed 19-June-2015]
- [PMLR14] PRYSS, R. ; MUNDBROD, N. ; LANGER, D. ; REICHERT, M.: Supporting Medical Ward Rounds Through Mobile Task and Process Management. In: *Information Systems and e-Business Management* (2014), Nr. 13, S. 107–146
- [PMR13] PRYSS, R. ; MUSIOL, S. ; REICHERT, M.: Collaboration Support Through Mobile Processes and Entailment Constraints. In: *Proc 9th IEEE Int'l Conf on Collaborative Computing: Networking, Applications and Worksharing*, IEEE Computer Society Press, 2013
- [PMR14] PRYSS, R. ; MUSIOL, S. ; REICHERT, M.: Extending Business Processes with Mobile Task Support: A Self-Healing Solution Architecture. In: *Handbook of Research on Architectural Trends in Service-Driven Computing* IGI Global, 2014, S. 99–130
- [PRBA15] PRYSS, R. ; REICHERT, M. ; BACHMEIER, A. ; ALBACH, J.: BPM to Go: Supporting Business Processes in a Mobile and Sensing World. In: *BPM Everywhere*. Future Strategies Inc., 2015
- [PRH⁺15] PRYSS, R. ; REICHERT, M. ; HERRMANN, J. ; LANGGUTH, B. ; SCHLEE, W.: Mobile Crowd Sensing in Clinical and Psychological Trials - A Case Study. In: *28th IEEE Int'l Symposium on Computer-Based Medical Systems*, IEEE Computer Society Press, June 2015

- [PRLS15] PRYSS, R. ; REICHERT, M. ; LANGGUTH, B. ; SCHLEE, W.: Mobile Crowd Sensing Services for Tinnitus Assessment, Therapy and Research. In: *IEEE 4th Int'l Conf on Mobile Services*, IEEE Computer Society Press, 2015
- [PRMD08] PHAM, Q. ; REID, J. ; MCCULLAGH, A. ; DAWSON, E.: Commitment Issues in Delegation Process. In: *Proc 6th Australian Conf on Information Security* Bd. 81 Australian Computer Society, 2008, S. 27–38
- [PS93] PIERCE, B. ; SANGIORGI, D.: Typing and Subtyping for Mobile Processes. In: *Proc of 8th Annual IEEE Symposium on Logic in Computer Science* IEEE, 1993, S. 376–385
- [PT06] POUSTTCHI, K. ; THURNHER, B.: Usage of Mobile Technologies to Support Business Processes. In: *Wireless Communication and Information* (2006), S. 101–120
- [PTKR10] PRYSS, R. ; TIEDEKEN, J. ; KREHER, U. ; REICHERT, M.: Towards Flexible Process Support on Mobile Devices. In: *Proc CAiSE'10 Forum*, Springer, 2010 (LNBIP 72), S. 150–165
- [PTR10] PRYSS, R. ; TIEDEKEN, J. ; REICHERT, M.: Managing Processes on Mobile Devices: The MARPLE Approach. In: *CAiSE'10 Demos*, 2010
- [Puc01] PUCELLA, R.: *The PI-Calculus: A Theory of Mobile Processes*. Cambridge, 2001
- [que14] *QuestionSys*. <http://www.uni-ulm.de/in/iui-dbis/forschung/projekte/questionsys.html>, 2014. – [Online; accessed 19-June-2015]
- [RA07] RUSSELL, N. ; AALST, W.M.P. van d.: Evaluation of the BPEL4People and WS-HumanTask Extensions to WS-BPEL 2.0 Using the Workflow Resource Patterns. In: *BPM Center Report BPM-07-10* (2007)
- [Rad12] RADEMAKERS, T.: *Activiti in Action: Executable Business Processes in BPMN 2.0*. Manning Publications Co., 2012
- [RB90a] REICHWALD, R. ; BESCHORNER, D.: Bewertung Benutzerfreundlicher Kommunikationssysteme unter Betriebswirtschaftlichen Aspekten. In: *Benutzerfreundliche Kommunikation* Bd. 15, Springer, 1990, S. 49–65
- [RB90b] REICHWALD, R. ; BESCHORNER, D.: Erweiterte Wirtschaftlichkeitsbetrachtung beim Einsatz neuer Kommunikationstechniken unter Berücksichtigung der VDI-Richtlinie 5015. In: *Dokumentation eines SAVE-Workshops* Bd. 15, Siemens-Informationstechnik-Anwenderverein, 1990, S. 49–65
- [RD97] REICHERT, M. ; DADAM, P.: A Framework for Dynamic Changes in Workflow Management Systems. In: *Proc 8th Int'l Workshop on Database and Expert Systems Applications* IEEE, 1997, S. 42–48
- [RD98] REICHERT, M. ; DADAM, P.: ADEPTflex—Supporting Dynamic Changes of Workflows Without Losing Control. In: *Journal of Intelligent Information Systems* 10 (1998), Nr. 2, S. 93–129
- [RD09] REICHERT, M. ; DADAM, P.: Enabling Adaptive Process-Aware Information Systems with ADEPT2. In: *Handbook of Research on Business Process Modeling*. Information Science Reference, 2009, S. 173–203

- [RDRM⁺09a] REICHERT, M. ; DADAM, P. ; RINDERLE-MA, S. ; JURISCH, M. ; KREHER, U. ; GOESER, K.: Architectural Principles and Components of Adaptive Process Management Technology. In: *PRIMIUM - Process Innovation for Enterprise Software*. Koellen-Verlag, 2009 (Lecture Notes in Informatics P-151), S. 81–97
- [RDRM⁺09b] REICHERT, M. ; DADAM, P. ; RINDERLE-MA, S. ; LANZ, A. ; PRYSS, R. ; PREDESCHLY, M. ; KOLB, J. ; LY, T. ; JURISCH, M. ; KREHER, U. ; GOESER, K.: Enabling Poka-Yoke Workflows with the AristaFlow BPM Suite. In: *Proc BPM'09 Demonstration Track*, 2009 (CEUR Workshop Proc 489)
- [Rei00] REICHERT, M.: Dynamische Ablaufänderungen in Workflow-Management-Systemen. In: *PhD Thesis, Ulm University* (2000)
- [RHD98] REICHERT, M. ; HENSINGER, C. ; DADAM, P.: Supporting Adaptive Workflows in Advanced Application Environments. In: *EDBT Workshop on Workflow Management Systems*, 1998, S. 100–109
- [Rin04] RINDERLE, S.: Schema Evolution in Process Management Systems. In: *Ulm University, PhD Thesis* (2004)
- [RKBB12] REICHERT, M. ; KOLB, J. ; BOBRIK, R. ; BAUER, T.: Enabling Personalized Visualization of Large Business Processes through Parameterizable Views. In: *27th ACM Symposium On Applied Computing*, ACM Press, 2012, S. 1653–1660
- [RLPL⁺13] RUF-LEUSCHNER, M. ; PRYSS, R. ; LIEBRECHT, M. ; SCHOBEL, J. ; SPYRIDOU, A. ; REICHERT, M. ; SCHAUER, M.: Preventing Further Trauma: KINDEX Mum Screen - Assessing and Reacting Towards Psychosocial Risk Factors in Pregnant Women with the Help of Smartphone Technologies. In: *XIII Congress of European Society of Traumatic Stress Studies*, 2013, S. 70–70
- [RM06] RECKER, J. ; MENDLING, J.: On the Translation Between BPMN and BPEL: Conceptual Mismatch Between Process Modeling Languages. In: *Proc 18th Int'l Conf on Advanced Information Systems Engineering* Namur University Press, 2006, S. 521–532
- [RMRW08] RINDERLE-MA, S. ; REICHERT, M. ; WEBER, B.: Relaxed Compliance Notions in Adaptive Process Management Systems. In: *Proc 27th Int'l Conference on Conceptual Modeling*, Springer, 2008 (LNCS 5231), S. 232–247
- [Rob11] ROBECKE, A.: Development of an iPhone Business Application. In: *Diploma Thesis, Ulm University* (2011)
- [ROM14] *ROME4U*. <http://www.tue.nl/en/publication/ep/p/d/ep-uid/274174/>, 2014. – [Online; accessed 19-June-2015]
- [RORCRC13] RÍO-ORTEGA, A. del ; RESINAS, M. ; CABANILLAS, C. ; RUIZ-CORTÉS, A.: On the Definition and Design-Time Analysis of Process Performance Indicators. In: *Information Systems* 38 (2013), Nr. 4, S. 470–490
- [RPR11] ROBECKE, A. ; PRYSS, R. ; REICHERT, M.: DBIScholar: An iPhone Application for Performing Citation Analyses. In: *CAiSE Forum-2011*, CEUR Workshop Proc, 2011 (Proc 23rd Int'l Conf on Advanced Information Systems Engineering Vol-73)

- [RR05] RINDERLE, S. ; REICHERT, M.: On the Controlled Evolution of Access Rules in Cooperative Information Systems. In: *Proc 13th Int'l Conf on Cooperative Information Systems*, Springer, 2005 (LNCS 3760), S. 238–255
- [RR06] RINDERLE, S. ; REICHERT, M.: Data-Driven Process Control and Exception Handling in Process Management Systems. In: *Proc 18th Int'l Conf on Advanced Information Systems Engineering*, Springer, 2006 (LNCS 4001), S. 273–287
- [RRD03] REICHERT, M. ; RINDERLE, S. ; DADAM, P.: On the Common Support of Workflow Type and Instance Changes under Correctness Constraints. In: *Proc 11th Int'l Conf Cooperative Information Systems*, Springer, 2003 (LNCS 2888), S. 407–425
- [RRD04] RINDERLE, S. ; REICHERT, M. ; DADAM, P.: Correctness Criteria for Dynamic Changes in Workflow Systems—A Survey. In: *Data & Knowledge Engineering* 50 (2004), Nr. 1, S. 9–34
- [RRHB09] REICHERT, M. ; RECHTENBACH, S. ; HALLERBACH, A. ; BAUER, T.: Extending a Business Process Modeling Tool with Process Configuration Facilities: The Provop Demonstrator. In: *Proc BPM'09 Demonstration Track*, 2009 (CEUR WS Proc 489)
- [RRKD05] REICHERT, M. ; RINDERLE, S. ; KREHER, U. ; DADAM, P.: Adaptive Process Management with ADEPT2. In: *Proc Int'l Conf on Data Engineering*, IEEE Computer Society Press, 2005, S. 1113–1114
- [RRMD03] REICHERT, R. ; RINDERLE-MA, S. ; DADAM, P.: ADEPT Workflow Management System Flexible Support for Enterprise-Wide Business Processes. In: *Proc 1st Int'l Conf on Business Process Management*, Springer, 2003 (LNCS 2678), S. 371–379
- [RS03] RITZ, T. ; STENDER, M.: Modellierung von Business-to-Business Geschäftsprozessen im Mobile Commerce. In: *Workshop Mobile Commerce*, 2003, S. 27–41
- [RS04] REICHERT, M. ; STOLL, D.: Komposition, Choreographie und Orchestrierung von Web Services: Ein Überblick. In: *EMISA Forum* 24 (2004), Nr. 2, S. 21–32
- [RSJ07] REIJERS, H.A. ; SONG, M. ; JEONG, B.: On the Performance of Workflow Processes with Distributed Actors: Does Place Matter? In: *Proc 5th Int'l Conf on Business Process Management*. Springer, 2007, S. 32–47
- [RW12] REICHERT, M. ; WEBER, B.: *Enabling Flexibility in Process-Aware Information Systems: Challenges, Methods, Technologies*. Springer, 2012
- [RWE14] *Responsive Webdesign*. http://de.wikipedia.org/wiki/Responsive_Webdesign, 2014. – [Online; accessed 19-June-2015]
- [SAM14] *SAMPROC*. <http://www-vs.informatik.uni-ulm.de/proj/samproc/>, 2014. – [Online; accessed 19-June-2015]
- [SAP14a] *Mobile Infrastructure*. http://de.wikipedia.org/wiki/Mobile_Infrastructure, 2014. – [Online; accessed 19-June-2015]

- [SAP14b] *SAP Business Process Management*. http://help.sap.com/saphelp_nwpi71/helpdata/de/8e/116642a3423654e10000000a155106/frameset.htm, 2014. – [Online; accessed 19-June-2015]
- [SAP14c] *SAP ERP*. http://de.wikipedia.org/wiki/SAP_ERP, 2014. – [Online; accessed 19-June-2015]
- [SAP14d] *SAP NetWeaver*. http://de.wikipedia.org/wiki/SAP_NetWeaver, 2014. – [Online; accessed 19-June-2015]
- [SARA04] SERRANO-ALVARADO, P. ; RONCANCIO, C. ; ADIBA, M.: A Survey of Mobile Transactions. In: *Distributed and Parallel Databases* 16 (2004), Nr. 2, S. 193–230
- [Sat96] SATYANARAYANAN, M.: Fundamental Challenges in Mobile Computing. In: *Proc 15th Annual ACM Symposium on Principles of Distributed Computing* ACM, 1996, S. 1–7
- [Sca14] *Scala*. [http://de.wikipedia.org/wiki/Scala_\(Programmiersprache\)](http://de.wikipedia.org/wiki/Scala_(Programmiersprache)), 2014. – [Online; accessed 19-June-2015]
- [SCFY96] SANDHU, R. ; COYNEK, E. ; FEINSTEINK, H. ; YOUMANK, C.: Role-Based Access Control Models yz. In: *IEEE Computer* 29 (1996), Nr. 2, S. 38–47
- [Sch09] SCHMIDT, H.: SAMProc - A Middleware for Highly Dynamic and Heterogeneous Environments. In: *PhD Thesis, Ulm University* (2009)
- [Sch11a] SCHNEIDERMEIER, T.: Usability of Mobile Devices. In: *Master Thesis, University Regensburg* (2011)
- [Sch11b] SCHOBEL, J.: Business Process Intelligence - Aktueller Stand und neue innovative Ansätze zur intelligenten Prozessanalyse. In: *Master Thesis, Ulm University* (2011)
- [Sch13] SCHINDLER, A.: Technische Konzeption und Realisierung des MACE-Tests mittels mobiler Technologie. In: *Bachelor Thesis, Ulm University* (2013)
- [Sco14] *Scopus*. <http://www.scopus.com/>, 2014. – [Online; accessed 19-June-2015]
- [SDGH09] SCHMIDT, H. ; DANG, C. ; GESSLER, S. ; HAUCK, F.: Model-Driven Development of Adaptive Applications with Self-Adaptive Mobile Processes. In: *Proc On the Move to Meaningful Internet Systems*. Springer, 2009, S. 726–743
- [SF03] SMITH, H. ; FINGAR, P.: *Business Process Management: The Third Wave*. Bd. 1. Meghan-Kiffer Press Tampa, 2003
- [SG94] SCHEER, A. ; GALLER, M.: Die Integration von Werkzeugen für das Management von Geschäftsprozessen. In: *Prozessorientierte Unternehmensmodellierung*. Springer, 1994, S. 101–118
- [SGDD07] SCHALL, D. ; GOMBOTZ, R. ; DORN, C. ; DUSTDAR, S.: Human Interactions in Dynamic Environments Through Mobile Web Services. In: *Proc IEEE Int'l Conf on Web Services* IEEE, 2007, S. 912–919
- [SGS⁺02] SAIRAMESH, J. ; GOH, S. ; STANOI, I. ; LI, C. ; PADMANABHAN, S.m: Self-Managing, Disconnected Processes and Mechanisms for Mobile E-Business. In: *Proc 2nd Int'l Workshop on Mobile Commerce* ACM, 2002, S. 82–89

- [SH06] SKOV, B. ; HÖGH, T.: Supporting Information Access in a Hospital Ward by a Context-aware Mobile Electronic Patient Record. In: *Personal and Ubiquitous Computing* 10 (2006), Nr. 4, S. 205–214
- [SHP⁺14a] SCHLEE, W. ; HERRMANN, J. ; PRYSS, R. ; REICHERT, M. ; LANGGUTH, B.: How Dynamic is the Continuous Tinnitus Percept? In: *11th Int'l Tinnitus Seminar*, 2014
- [SHP⁺14b] SCHLEE, W. ; HERRMANN, J. ; PRYSS, R. ; REICHERT, M. ; LANGGUTH, B.: Moment-to-Moment Variability of the Auditory Phantom Perception in Chronic Tinnitus. In: *13th Int'l Conf on Cochlear Implants and Other Implantable Auditory Technologies*, 2014
- [Sin07] SINISI, J.: *System and Method for Mobile Data Collection*. 2007. – US Patent 7,313,759
- [SKH07] SCHMIDT, H. ; KAPITZA, R. ; HAUCK, F.: Mobile-Process-Based Ubiquitous Computing Platform: A Blueprint. In: *Proc 1st Workshop on Middleware-Application Interaction* ACM, 2007, S. 25–30
- [SKHR08] SCHMIDT, H. ; KAPITZA, R. ; HAUCK, F. ; REISER, H.: Adaptive Web Service Migration. In: *Distributed Applications and Interoperable Systems* Springer, 2008, S. 182–195
- [SKR14] SEMMELRODT, F. ; KNUPLESCH, D. ; REICHERT, M.: Modeling the Resource Perspective of Business Process Compliance Rules with the Extended Compliance Rule Graph. In: *15th Int'l Working Conf on Business Process Modeling, Development, and Support*, Springer, 2014 (LNBIP 175), S. 48–63
- [SLS06] SCHAAD, A. ; LOTZ, V. ; SOHR, K.: A Model-Checking Approach to Analysing Organisational Controls in a Loan Origination Process. In: *Proc 11th ACM Symposium on Access Control Models and Technologies* ACM, 2006, S. 139–149
- [SM02] SCHAAD, A. ; MOFFETT, J.: A Framework for Organisational Control Principles. In: *Proc 18th Conf on Annual Computer Security Applications* IEEE, 2002, S. 229–238
- [Sma14] *Smartphone*. <http://de.wikipedia.org/wiki/Smartphone>, 2014. – [Online; accessed 19-June-2015]
- [SPR15] SCHOBEL, J. ; PRYSS, R. ; REICHERT, M.: Using Smart Mobile Devices for Collecting Structured Data in Clinical Trials: Results From a Large-Scale Case Study. In: *28th IEEE Int'l Symposium on Computer-Based Medical Systems*, IEEE Computer Society Press, 2015
- [SPSR15] SCHICKLER, M. ; PRYSS, R. ; SCHOBEL, J. ; REICHERT, M.: An Engine Enabling Location-based Mobile Augmented Reality Applications. In: *10th Int'l Conf on Web Information Systems and Technologies - Revised Selected Papers*. Springer, 2015 (LNBIP)
- [SQL14a] *SQLData-Java-Interface*. <http://docs.oracle.com/javase/7/docs/api/java/sql/SQLData.html>, 2014. – [Online; accessed 19-June-2015]

- [SQL14b] *SQLData-Java-Interface II*. <http://www.kbs.twi.tudelft.nl/Documentation/Programming/Java/jdk1.2/guide/jdbc/spec2/jdbc2.0.frame9.html>, 2014. – [Online; accessed 19-June-2015]
- [SRG08] SEN, R. ; ROMAN, G. ; GILL, C.: CIAN: A Workflow Engine for Manets. In: *Coordination Models and Languages* Springer, 2008, S. 280–295
- [SRLP⁺13] SCHOBEL, J. ; RUF-LEUSCHNER, M. ; PRYSS, R. ; REICHERT, M. ; SCHICKLER, M. ; SCHAUER, M. ; WEIERSTALL, R. ; ISELE, D. ; NANDI, C. ; ELBERT, T.: A Generic Questionnaire Framework Supporting Psychological Studies with Smartphone Technologies. In: *XIII Congress of European Society of Traumatic Stress Studies*, 2013, S. 69–69
- [SRP⁺15] SCHICKLER, M. ; REICHERT, M. ; PRYSS, R. ; SCHOBEL, J. ; LANGGUTH, B. ; SCHLEE, W.: *Entwicklung mobiler Apps: Konzepte, Anwendungsbausteine und Werkzeuge im Business und E-Health*. Springer, 2015
- [SSP⁺13] SCHOBEL, J. ; SCHICKLER, M. ; PRYSS, R. ; NIENHAUS, H. ; REICHERT, M.: Using Vital Sensors in Mobile Healthcare Business Applications: Challenges, Examples, Lessons Learned. In: *9th Int'l Conf on Web Information Systems and Technologies*, 2013, S. 509–518
- [SSP⁺14] SCHOBEL, J. ; SCHICKLER, M. ; PRYSS, R. ; MAIER, F. ; REICHERT, M.: Towards Process-Driven Mobile Data Collection Applications: Requirements, Challenges, Lessons Learned. In: *10th Int'l Conf on Web Information Systems and Technologies*, 2014, S. 371–382
- [SSPR15] SCHOBEL, J. ; SCHICKLER, M. ; PRYSS, R. ; REICHERT, M.: Process-Driven Data Collection with Smart Mobile Devices. In: *10th Int'l Conf on Web Information Systems and Technologies - Revised Selected Papers*. Springer, 2015 (LNBIP)
- [Sta14] STACH, M.: Konzeption und Realisierung einer mobilen Schnittstelle gegen ein aktuelles Enterprise Content-Management-System. In: *Bachelor Thesis, Ulm University* (2014)
- [Sto08] STOIMENOVA, B.: Evaluation von Workflow-Sprachen zur Prozessmodellierung in Multi-Domain Umgebungen am Beispiel von Géant2 E2E Links. In: *Munich Technical University, Master Thesis* (2008)
- [Sto10] STOICSICS, M.: Anforderungsanalyse Airline Catering mittels BPMN-Prozessen. In: *Student Project, Ulm University* (2010)
- [Sto11] STOICSICS, M.: Evaluation des Konzeptes eines Catering-, Steuerungs- und Controllingsystems am Vergleichsbeispiel des Endmontageprozesses der Fertigung von hochisolierenden Lager- und Transportbehältern für das Luftfahrt-Catering. In: *Diploma Thesis, Ulm University* (2011)
- [STo14] *Sencha Touch*. http://de.wikipedia.org/wiki/Sencha_Touch, 2014. – [Online; accessed 19-June-2015]
- [SUP14] *Sybase Unwired Platform*. http://en.wikipedia.org/wiki/Sybase_Unwired_Platform, 2014. – [Online; accessed 19-June-2015]
- [SW03] SANGIORGI, D. ; WALKER, D.: *The Pi-Calculus: A Theory of Mobile Processes*. Cambridge University Press, 2003

- [Tab14] *Tabletcomputer*. <http://de.wikipedia.org/wiki/Tabletcomputer>, 2014. – [Online; accessed 19-June-2015]
- [Thr14] *Thread*. [http://de.wikipedia.org/wiki/Thread_\(Informatik\)](http://de.wikipedia.org/wiki/Thread_(Informatik)), 2014. – [Online; accessed 19-June-2015]
- [UELW10] UNGER, T. ; EBERLE, H. ; LEYMAN, F. ; WAGNER, S.: An Event-Model for Constraint-Based Person-Centric Flows. In: *Int'l Conf on Progress in Informatics and Computing* Bd. 2 IEEE, 2010, S. 927–932
- [UMT14] *Universal Mobile Telecommunications System*. <http://de.wikipedia.org/wiki/Umts>, 2014. – [Online; accessed 19-June-2015]
- [Use13] USELMANN, A.: Enterprise Mobility mit der SAP Mobile Infrastructure: Untersuchung der Sybase Unwired Platform anhand einer Fallstudie im Bereich Instandhaltung unter Einbezug geografischer Daten. In: *Diploma Thesis, Ulm University* (2013)
- [VAP14] *Vier-Augen-Prinzip*. <http://de.wikipedia.org/wiki/Vier-Augen-Prinzip>, 2014. – [Online; accessed 19-June-2015]
- [VO02] VENTER, K. ; OLIVIER, M.: The Delegation Authorization Model: A Model for the Dynamic Delegation of Authorization Rights in a Secure Workflow Management System. In: *ISSA*, 2002, S. 1–13
- [WC02] WIEDEMUTH-CATRINESCU, U.: Evolution von Organisationsmodellen in Workflow-Management-Systemen. In: *Diploma Thesis, Ulm University* (2002)
- [WCK11] WAKHOLI, P. ; CHEN, W. ; KLUNGSØYR, J.: Workflow Support for Mobile Data Collection. In: *Enterprise, Business-Process and Information Systems Modeling*. Springer, 2011, S. 299–313
- [WCL⁺05] WEERAWARANA, S. ; CURBERA, F. ; LEYMAN, F. ; STOREY, T. ; FERGUSON, D.: *Web Services Platform Architecture: SOAP, WSDL, WS-Policy, WS-Addressing, WS-BPEL, WS-Reliable Messaging and More*. Prentice Hall PTR, 2005
- [Web14] *Web Content Management System*. http://en.wikipedia.org/wiki/Web_content_management_system, 2014. – [Online; accessed 19-June-2015]
- [Wes07] WESKE, M.: *Business Process Management: Concepts, Languages, Architectures*. Bd. 14. Springer, 2007
- [Wie14] WIERINGA, R.: *Design Science Methodology for Information Systems and Software Engineering*. Springer, 2014
- [win14] *Windows Phone*. http://de.wikipedia.org/wiki/Microsoft_Windows_Phone, 2014. – [Online; accessed 19-June-2015]
- [WJH97] WARD, A. ; JONES, A. ; HOPPER, A.: A New Location Technique for the Active Office. In: *IEEE Personal Communications* 4 (1997), Nr. 5, S. 42–47
- [WKB07] WAINER, J. ; KUMAR, A. ; BARTHELMESS, P.: DW-RBAC: A Formal Security Model of Delegation and Revocation in Workflow Systems. In: *Information Systems* 32 (2007), Nr. 3, S. 365–384

- [WMC94] *Workflow Reference Model (WFMC-TC-1003, 29-Nov-94, 1.1)*. <ftp://ftp.aiai.ed.ac.uk/pub/projects/WfMC/refmodel/rmv1-16.pdf>, 1994. – [Online; accessed 19-June-2015]
- [WMM09] WOLTER, C. ; MISELDINE, P. ; MEINEL, C.: Verification of Business Process Entailment Constraints Using SPIN. In: *Engineering Secure Software and Systems*. Springer, 2009, S. 1–15
- [Woh12] WOHLHAUPTER, P.: Research in Business Process Management: A Bibliometric Analysis. In: *Diploma Thesis, Ulm University* (2012)
- [WPA14] *Software-Agent*. <http://de.wikipedia.org/wiki/Software-Agent>, 2014. – [Online; accessed 19-June-2015]
- [WRRM08] WEBER, B. ; REICHERT, M. ; RINDERLE-MA, S.: Change Patterns and Change Support Features - Enhancing Flexibility in Process-Aware Information Systems. In: *Data and Knowledge Engineering* 66 (2008), Nr. 3, S. 438–466
- [WS07] WOLTER, C. ; SCHAAD, A.: Modeling of Task-based Authorization Constraints in BPMN. In: *5th Int'l Conf on Business Process Management*. Springer, 2007, S. 64–79
- [WSC14] *WS-Choreography Description Language*. http://de.wikipedia.org/wiki/WS-Choreography_Description_Language, 2014. – [Online; accessed 19-June-2015]
- [WSH14] *WS-HumanTask*. <http://de.wikipedia.org/wiki/WS-HumanTask>, 2014. – [Online; accessed 19-June-2015]
- [WSM08] WOLTER, C. ; SCHAAD, A. ; MEINEL, C.: Task-based Entailment Constraints for Basic Workflow Patterns. In: *Proc 13th ACM Symposium on Access Control Models and Technologies* ACM, 2008, S. 51–60
- [WSNL07] WEBER, H. ; STÖCKLI, M. ; NÜBLING, M. ; LANGEWITZ, W.: Communication During Ward Rounds in Internal Medicine: An Analysis of Patient–Nurse–Physician Interactions Using RIAS. In: *Patient Education and Counseling* 67 (2007), Nr. 3, S. 343–348
- [WWLR06] WEBER, B. ; WILD, W. ; LAUER, M. ; REICHERT, M.: Improving Exception Handling by Discovering Change Dependencies in Adaptive Process Management Systems. In: *Proc 2nd Int'l Workshop on Business Process Intelligence*, Springer, 2006 (LNCS 4103), S. 93–104
- [YAEB09] YILDIZ, M. ; ABAWAJY, J. ; ERCAN, T. ; BERNOTH, A.: A Layered Security Approach for Cloud Computing Infrastructures. In: *10th Int'l Symposium on Pervasive Systems, Algorithms, and Networks* IEEE, 2009, S. 763–767
- [YLD+12] YANG, Y. ; LU, W. ; DOMACK, J. ; LI, T. ; CHEN, S. ; LUIS, S. ; NAVLAKHA, J.: MADIS: A Multimedia-Aided Disaster Information Integration System for Emergency Management. In: *8th Int'l Conf on Collaborative Computing: Networking, Applications and Worksharing*, 2012, S. 233–241
- [Zap05] ZAPLATA, S.: Prozessintegration in Middleware für Mobile Systeme. In: *Diploma Thesis, University of Hamburg* (2005)

- [ZDL09] ZAPLATA, S. ; DREILING, V. ; LAMERSDORF, W.: Realizing Mobile Web Services for Dynamic Applications. In: *Software Services for e-Business and e-Society*. Springer, 2009, S. 240–254
- [ZHKL10] ZAPLATA, S. ; HAMANN, K. ; KOTTKE, K. ; LAMERSDORF, W.: Flexible Execution of Distributed Business Processes Based on Process Instance Migration. In: *Journal of Systems Integration* 1 (2010), Nr. 3, S. 3–16
- [ZL10] ZAPLATA, S. ; LAMERSDORF, W.: Towards Mobile Process as a Service. In: *Proc ACM Symposium on Applied Computing* Bd. 1, ACM, 2010, S. 372–379

Tabellenverzeichnis

1.1	Leitlinien des <i>Design Science</i> -Ansatzes und deren Umsetzung	11
2.1	Suchbegriffmenge der Literaturrecherche	14
2.2	Erweiterte Suchbegriffmenge der Literaturrecherche	14
2.3	Literaturrecherche mittels Suchmaschinen und BibTechMon-Analyse	16
2.4	Fallstudien	18
2.5	Auswahl typischer Szenarien zur Klinischen Visite	20
2.6	Identifizierte Datenzugriffe und Zugriffsrechte	22
2.7	Anforderungen an eine robuste Ausführung mobiler Aktivitäten	26
3.1	Wichtige Aspekte der Prozessumgebung	30
3.2	Wichtige Aspekte des mobilen Prozess-Klienten	31
3.3	Vor- und Nachteile mobiler Anwendungsentwicklung im Überblick	38
3.4	Forschungsprojekte im Kontext mobiler Anwendungsentwicklung	39
3.5	Vergleich der Konzepte mobiler Anwendungsentwicklung	40
3.6	Sichtweisen und Prozessbeispiele	43
4.1	Kategorien verwandter Ansätze	46
4.2	Anforderungsübersicht dieser Arbeit und Stand der Technik	53
4.3	Anmerkungen zu den Anforderungen	54
5.1	Prozess-Metamodell und mobile Aktivitäten	60
5.2	Verwendete Endanwenderausdrücke (vereinfacht)	66
5.3	Auswertungsvarianten für Endanwenderausdrücke	68
6.1	Beschreibung der Architekturkomponenten	73
6.2	Prozess-Klient und Aktivitätenausführung	75
6.3	Worklist-Operationen	77
6.4	Phasen des Ausführungsprotokolls	78
7.1	Parameter-Übersicht	91
7.2	Methoden der Online-Status Ermittlung eines mobilen Endanwenders	97
7.3	Ermittlungsvarianten des Online-Status mobiler Endanwender	98
7.4	Parameter und ihr Spezifikations- und Verwaltungsaufwand	103
8.1	Parametermodellierung	106
8.2	Kontextregeln für Ranking-Algorithmus	111
8.3	Kontextparameter zur Ausführung mobiler Aktivitäten	116
8.4	Übertragene Parameter	120
8.5	Beitrag zur robusten Ausführung mobiler Aktivitäten	121
9.1	Fragestellungen im Kontext der Delegation mobiler Aktivitäten	127
9.2	Kontextregeln zur Delegationsaktivierung	128
9.3	Kontextregeln für Delegationen	129
9.4	Fallunterscheidung Smart-Mobilgerät	131

9.5	Phasen des Ausführungsprotokolls bei Delegationen	136
9.6	Phasen des Ausführungsprotokolls bei Zwischenspeicherung von Daten	138
9.7	Delegationsvarianten	138
9.8	Backupaspekte mobiler Aktivitäten	140
9.9	Backupvarianten	141
9.10	Übertragene Parameter	148
9.11	Beitrag der Ausnahmebehandlung zur robusten Ausführung mobiler Aktivitäten	149
10.1	Prozess-Constraints für Endanwender (vereinfacht)	152
10.2	Praxisrelevanz von Prozess-Constraints	153
10.3	Praxisrelevanz einer kombinierten Anwendung von Prozess-Constraints	155
10.4	Anpassungen für Prozess-Constraints	157
10.5	Anpassungen für kombinierte Anwendung der Prozess-Constraints	161
10.6	Anpassungen für die Prozess-Constraints bei der Ausnahmebehandlung	162
10.7	Anpassungen für kombinierte Prozess-Constraints bei der Ausnahmebehandlung	173
10.8	Beitrag der Prozess-Constraints zur robusten Ausführung mobiler Aktivitäten .	174
11.1	Erläuterungen zu den Varianten einer mobilen Prozess-Management-Architektur	176
12.1	REST-Schnittstelle zur Autorisierung von mobilen Endanwendern	193
13.1	Einsatz des Prototyps in Fallstudien	202
13.2	Mobile Aktivitäten zur Fallstudie in der Lagerverwaltung	203
13.3	Mobile Aktivitäten zur Fallstudie im Rettungsdienst	204
13.4	Mobile Aktivitäten zur Fallstudie Klinischer Studien	205

Abbildungsverzeichnis

1.1	Klinikprozess einer Röntgenuntersuchung in BPMN-Notation	8
1.2	Aufbau der Arbeit	12
2.1	BibTechMon-Analyse Sub-Phase 1	15
2.2	BibTechMon-Analyse Sub-Phase 2	16
2.3	Angewandte Methodik	18
2.4	Beteiligte Klinische Abteilungen in Fallstudie 1	19
2.5	Visiteprozess der Inneren Medizin	20
2.6	Prozess der Röntgenuntersuchung	21
2.7	Informationsarchitektur des mobilen Prototyps der Klinischen Visite	22
2.8	Prozess zur Phase 5 der Kommissionierung	23
2.9	Prozess zwischen Klinik, Patient und Gesundheitsdienstleister	24
3.1	Automatische und interaktive mobile Aktivitäten	32
3.2	Orchestrierung und Choreographie mobiler Prozesse	33
3.3	Fragmentierung und Instanz-Migration	34
3.4	Nativer und Browser-basierter mobiler Prozess-Klient	40
3.5	Ad-hoc Änderungen mobiler Aktivitäten	42
3.6	Beispielprozesse verschiedener Anwendungsdomänen	43
4.1	Flow-Design mit der SAP Sybase Unwired Platform [Use13]	52
4.2	IBM: Markieren einer mobilen Aktivität [IBM14c]	52
4.3	MARPLE Mediation Center	53
5.1	Prozess-Lebenszyklus	58
5.2	Prozess-Metamodell (vereinfacht)	59
5.3	ADEPT-Prozessmodell (vereinfacht)	61
5.4	Darstellung bedingter Verzweigungen in Prozessen	61
5.5	Statechart einer Aktivitäteninstanz (vereinfacht)	62
5.6	Darstellung von Aktivitätszuständen in Prozessen	63
5.7	Darstellung von Kantenzuständen in Prozessen	63
5.8	Prozessbeispiel mit Aktivitäten- und Kantenzuständen	64
5.9	Darstellung des Datenflusses in Prozessen	64
5.10	Darstellung von Prozess-Constraints	67
5.11	Nicht erlaubte Anwendung von Separation of Duties	67
5.12	Prozessinstanz mit Ausführungszuständen	70
6.1	Architekturkomponenten im Überblick	72
6.2	Prozess-Klient	74
6.3	Worklist-Beispiele (vereinfacht)	76
6.4	Protokoll der Aktivitätsausführung	79
7.1	Prozessfragment mit mobiler Aktivität	84
7.2	Angewandte Methodik für mobilen Kontext	87

7.3	Verknüpfungsarten mobiler Endanwender mit Smart-Mobilgeräten	89
7.4	Prozessfragment mit festgelegtem Ausführungsort für mobile Aktivität	93
7.5	Einzugsbereich und Ausführungsort einer mobilen Aktivität	94
7.6	Funktionale Abhängigkeit von MA_{OFF} und MA_D	95
7.7	Modellierung und Verwaltung des mobilen Kontexts in der <i>PrMS</i> -Architektur .	101
8.1	Erweiterung des Meta-Modells für mobile Aktivitäten	107
8.2	Beispielberechnungen für $OT(n)$	110
8.3	Beispielauswertungen des primären <i>Ranking-Algorithmus</i>	113
8.4	Systemwechsel	117
8.5	Berücksichtigung von Prozess-Strukturen	119
8.6	Ausführungsprotokoll und Parameterübertragung	120
9.1	Ausnahmebehandlungsstrategie mobiler Aktivitäten	125
9.2	Erweitertes Zustandsmodell (vereinfacht)	126
9.3	Prozess-Klient mit verursachter Delegation	132
9.4	Prozess-Klient mit angebotener Delegation	133
9.5	Betroffener Bereich des Ausführungsprotokolls bei Ausnahmen	134
9.6	Ausführungsprotokoll und Delegationen	135
9.7	Ausführungsprotokoll und Zwischenspeicherung von Daten	137
9.8	Erweitertes Zustandsmodell (vereinfacht)	138
9.9	Backupvariante BV1: Backup	142
9.10	Backupvariante BV2: Beschleunigtes Backup	143
9.11	Prozess-Klient und Backup-Zustand	144
9.12	Notwendige Listenberechnungen für Endanwender der Backupvariante BV1 . .	145
9.13	Notwendige Listenberechnungen für Endanwender der Backupvariante BV2 . .	145
9.14	Parameterübertragung	147
9.15	Prozessinstanz in Ausführung mit Ausnahmebehandlung	148
10.1	Kombinierte Anwendung von Prozess-Constraints	154
10.2	Kombinierte Anwendung von Card- und BoD-Constraints	154
10.3	Kombinierte Anwendung mehrerer BoD-Constraints (BoD-Kette)	156
10.4	Binding of Duties und Worklist	159
10.5	Binding of Duties mit dazwischenliegender mobiler Aktivität	160
10.6	Ausnahmebehandlung bei Vorliegen eines <i>SoD</i> -Constraints	163
10.7	Delegationslistenanpassung bei kombiniertem <i>SoD</i> -Constraint	167
10.8	Prozess-Constraints und nicht mobile Aktivitäten	167
10.9	Weitere Prozess-Constraints im Kontext mobiler Aktivitäten	168
10.10	Kombinierte Anwendung der Konzepte	169
11.1	Varianten für eine mobile Prozess-Management-Architektur	176
11.2	Architekturkomponenten mobiler Prozess-Klient	177
11.3	Architekturkomponenten einer mobilen Aktivitätsausführung	180
11.4	Session-Handling für mobile Endanwender	181
11.5	Architekturkomponenten für Prozess-Modellierung	184
11.6	Informationsarchitektur mobiler Prozess-Klient	185
12.1	Konzeption des Prototyps	190

12.2	Integrationsablauf zwischen Prototyp und PrMS	190
12.3	Service-orientierte Middleware	191
12.4	Verwendungszweck der Python-Skripte	192
12.5	Zuteilung mobiler Aktivitäten	194
12.6	Ausführung mobiler Aktivitäten	195
12.7	Beenden einer mobilen Aktivität	196
12.8	Repository mit gespeicherten Daten	196
12.9	Mobiler Prozess-Klient auf Android	197
12.10	Mobiler Prozess-Klient auf iOS	198
12.11	Konfiguration mobiler Aktivitäten mithilfe von 3D-Modellen [PRBA15]	199
13.1	Fragebogen auf Smart-Mobilgerät	206
14.1	Rahmenwerk zur robusten Ausführung mobiler Aktivitäten in Prozessumgebungen	211